Public Key-Enabled Application Family of Protection Profiles

(Individual PP Names are defined by the algorithm:

USMC PKE PP with <packages included in the PP, listed in the order in which they appear in the PP> **at EAL** <3 or 4, depending on the EAL selected> **with augmentation**

and are discussed further in the Foreword)

Version 2.5

Date: October 31, 2002

Prepared for: US Marine Corps

Foreword

This family of PPs is written to support the development of a range of Public Key-Enabled applications and services that may be integrated into a computing platform. This family of PPs is written to address applications and products written for and used in the United States Department of Defense.

Given the range of applications to which it may be applied, the approach used in writing this family of PPs was to use the concept 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. Packages may be thought of as sets of defined functionality requirements. All PKE applications are required to perform certain processes. Other processes may or may not be performed, depending upon the needs and functions of the application.

A base set of functional requirements was defined that must be met by all PKE applications compliant with the PP. In addition, packages were defined that contain functionality that may or may not be included in a PKE application. The functionality contained in the packages is not "optional." Rather, the packages define additional PK functionality that may or may not be needed by an application (TOE). If a particular application (TOE) contains the functionality defined in a given package, then that package must be included in the ST for the TOE and the TOE must comply with the package requirements in full.

In addition, this family of PPs contains two different Assurance Levels (EALs). The appropriate EAL will be selected by the ST author depending upon the requirements of the application.

Thousands of possible PPs are included in this PP family, given the number of possible combinations of packages and the choice of EAL. Rather than listing thousands of names, an algorithm was defined to generate the name of any given PP. The PP name is of the form:

> **USMC PKE PP with** <packages included in the PP, listed in the order in which they appear in the PP> **at EAL** <3 or 4, depending on the EAL selected> **with augmentation**

The words in bold print are included in every title, appropriate package names are listed for all of the packages included in the PP, and the EAL chosen is specified. Note that the list of packages in the title must be in the order in which they appear in this document in order to ensure consistency of naming.

Revision History

Version	Date	Description
0.1	March 15, 2002	Initial version of the Protection Profile
0.2	March 22, 2002	Draft version of the Protection Profile. Added threats, objectives, and requirements.
0.3	April 6, 2002	Added a new PP for PKI Based Entity Authentication Revised the PKI Credential Management package for additional security requirements Updated the Glossary to make the definitions more accurate Responded to comments received
1.0	April 30, 2002	Made editorial changes including cleaning up, adding references, expanding acronyms, adding acronyms list, etc. Revised the CC assurance requirements based on comments received. Provided for dependencies for cryptographic operations on extended requirements, as appropriate (e.g., FCS_COP.1 when a cryptographic operation is involved) Added optional trust anchor processing Added optional Audit functional package Revised Section 2 to provide a better approach to the reader regarding how to read the document. Added a new subsection “Approach” Revised threats and objectives based on comments received
1.1	May 28, 2002	Revised to explain assumption regarding the functional Vs. procedural aspects of path validation. Added an assumption regarding how the key recovery is out of scope Added the approach how multiple keys (e.g., due to key recovery, key history, re-key, etc.) are supported Made changes based on NIST comments, including: ▪ Added ALC_FLR.1 requirements ▪ Improved explanation of FIPS 140 series requirements in terms requiring it for all cryptographic modules ▪ Made the path validation packages incremental as opposed to self-contained ▪ Explained the path validation packages better ▪ Added rules for rejecting certificates, CRL, OCSP responses if critical extensions are not processed by the TOE ▪ Made minor revisions to the threats and objectives ▪ Made minor changes to SFR

Version	Date	Description
1.2	June 16, 2002	Responded to evaluator EORs and authors reviewed changes.
1.3	June 28, 2002	Responded to additional EORs.
1.3.1	July 1, 2002	Fixed type “envelop” to “envelope” in table 2.1
2.0	July 25, 2002	Responded to EORs.
2.1	August 1, 2002	Separated document into 28 PPs at the direction of NIAP.
2.2	September 22, 2002	Responded to decisions by NIAP regarding the use of packages.
2.3	September 26, 2002	Made minor updates to fix errors in previous version.
2.4	October 25, 2002	Responded to EORs: EOR_APE_DES.1-04, EOR_APE_DES.1-05, EOR_APE_DES.1.06, EOR_APE_ENV.1-02, EOR_APE_ENV.1-03, EOR_APE_OBJ.1-01, EOR_APE_OBJ.1-02, EOR_APE_OBJ.1-03, EOR_APE_REQ.1-06, EOR_APE_REQ.1-14, EOR_APE_REQ.1-15, EOR_APE_REQ.1-16, EOR_APE_REQ.1-17, EOR_Editorial_PP_v2-3
2.5	October 31, 2002	Responded to evaluator questions and revised the PP for editorial changes. No major material change was made.

Table of Contents

Page

List of Tables

Page

1 Introduction

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

1.1 Identification

Title: Thousands of possible PPs are included in this PP family, given the number of possible combinations of packages and the choice of EAL. Rather than listing the names, an algorithm was defined to generate the name of any given PP. The PP name is of the form:

> **USMC PKE PP with** <packages included in the PP, listed in the order in which they appear in the PP> **at EAL** <3 or 4, depending on the EAL selected> **with augmentation**

The words in bold print are included in every title, appropriate package names are listed for all of the packages included in the PP, and the EAL chosen is specified. Note that the list of packages in the title must be in the order in which they appear in this document in order to ensure consistency of naming.

Assurance Level: This family of PPs includes Evaluation Assurance Level (EAL) of EAL 3 with Augmentation and EAL 4 with Augmentation. The functional requirements, objectives, threats, and assumptions are identical for each EAL. The ST author will choose the appropriate EAL depending upon the needs of the application. The Strength of Function (SOF) in all PPs is SOF Basic.

Version Number: Version 2.5

Date: October 31, 2002

PP Authors: Jean Petty, CygnaCom Solutions, Inc and Santosh Chokhani, Orion Security Solutions, Inc.

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 Protection Profile Overview

This family of PPs describes the Information Technology (IT) security requirements for PKE Applications, based on the X.509 standard (see references below), integrated into computing platforms or systems. Public key technology provides digital signature generation and verification, public/private key encryption and decryption, public key distribution services, and various support functions. A PKE application may provide confidentiality, integrity, authentication, and non-repudiation, based on the use of public key technology security services. A variety of applications may be PK-enabled. This family of PPs defines different PK services. Thousands of possible PPs may be defined

depending upon the combination of functional packages and the EAL chosen to meet the requirements of the application. Many functional requirements in the PPs represent extensions to the Common Criteria (CC), because the CC does not provide requirements for the X.509 processing rules that are critical to this family of PPs.

1.3 Related Documents

- Department of Defense (DoD) Class 3 Public Key Infrastructure (PKI) Public Key-Enabled Application Requirements," Version 1.0, 13 July 2000
- DoD Public Key Infrastructure Interoperability Master Test Plan, Version 1.2, dated November 2001.
- DoD Class 3 PKI Concept of Operations (CONOPS), National Security Agency (NSA) Security Evaluation Version, 10 November 1999
- Federal Information Processing Standard (FIPS) 196, Entity Authentication Using Public Key Cryptography, 18 February 1997
- International Organization for Standards/Internet Electrotechnical Committee (ISO/IEC) 9594-8:"Information Technology- Open Systems Interconnection-The Directory: Public Key and Attribute Certificate Frameworks" (X.509 Standard)
- X.509 Certificate Policy for the United States DoD, Version 5.0, 13 December 1999
- X.509 Internet Public Key Infrastructure Certificate and CRL Profile, RFC 2459, January 1999
- X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP) <draft-ietf-pkix-rfc2560bis-01.txt>, February 2002
- International Standard ISO/IEC 15408 Information technology — Security techniques — Evaluation criteria for IT security
- Common Methodology for Information Security Evaluation (CEM) Version 1.0, August 1999
- FIPS 140-2, Security Requirements for Cryptographic Modules, 25 May 2001 http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

1.4 PP Organization

This family of PPs includes thousands of possible PPs that describe different PK services. Sections 2, 3, and 4 define TOE descriptions, assumptions and threats, and security objectives, respectively. The descriptions, assumptions and threats, and security objectives are identified separately for each package defined. Sections 5.1 through 5.3 provide the security functional requirements for all of the packages. Sections 5.4 and 5.5 describe EAL 3 augmented and EAL 4 augmented requirements, respectively. Either EAL 3 with augmentation or EAL 4 with augmentation will be selected depending on the requirements of the application. Rationale is included in Section 6.

A glossary of PKI-related terms used in the protection profile (PP) is provided in the Appendix followed by a list of acronyms.

1.5 Common Criteria Conformance

This family of PPs has been built with Common Criteria (CC) Version 2.1 (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).

The PPs at assurance level EAL 3 augmented in this family of PPs are Common Criteria Version 2.1, Part 2 extended, and Part 3 conformant, at Evaluation Assurance Level 3 with Augmentation. The PPs at assurance level EAL 4 augmented in this family of PPs are Common Criteria Version 2.1, 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 3, section 5.4, “Part 2 extended - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2.”

2 TOE Description

2.1 Overview

An application is PK enabled if it:

- Securely manages keys, trust anchors, and certificates.
- Uses one or more of the security services supported by the DoD PKI by accepting and processing a DoD X.509 digital certificate.
- Is able to obtain relevant certificates and revocation data.
- Checks each certificate for validity, using procedures described in the X.509 standard [ISO 9594-8], prior to reliance, including checking for revocation.
- Has access to accurate and reliable time in order to verify the dates on certificates, revocation data, and application data.
- Correctly interoperates with the Common Access Card (CAC) or another DoD approved "hard token".
- Collects, stores and maintains the data required to support digital signature verification in the future.
- Is able to automatically select from multiple private decryption keys if it performs public key based decryption.

A PKE application must interoperate “correctly” with the DoD PKI. The Defense Information Systems Agency (DISA), Joint Interoperability Test Center (JITC) has developed the “Department of Defense Public Key Infrastructure Interoperability Master Test Plan Version 1.2, dated November 2001”. DISA has determined that a PK-Enabled application interoperates “correctly” with the DoD PKI if the application successfully completes this test protocol. In this family of PPs, application is synonymous with Target of Evaluation (TOE).

2.2 Approach

This section defines the approach that was taken in developing this family of PPs. This document does not provide background information on the CC, PKI, PKE, or cryptography. The reader is assumed to possess appropriate knowledge of the CC, PKI, cryptography and related technology to understand the content of this document. There are, however, many ways to develop a PP and to address the subject matter of this family of PPs. This section provides specifics on the development approach used.

2.2.1 Package concept

This PP family provides a tool to specify and evaluate a broad range of PKE applications. Given the range of applications to which it may be applied, the approach used in writing this PP family was to use the concept 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. Packages may be thought of as sets of defined functionality requirements. All PKE

applications are required to perform certain processes. Other processes may or may not be performed, depending upon the needs and functions of the application.

A base set of functional requirements was defined that must be met by all PKE applications compliant with any PP in this family of PPs. In addition, packages were defined that contain functionality that may or may not be included in a PKE application. The functionality contained in the packages is not "optional." Rather, the packages define additional PK functionality that may or may not be needed by an application (TOE). If a particular application (TOE) contains the functionality defined in a given package, then that package must be included in the ST for the TOE and the TOE must comply with the package requirements in full. Thousands of possible PPs are included in this PP family, given the number of possible combinations of packages and the choice of EAL. Rather than listing thousands of names, an algorithm was defined to generate the name of any given PP. The PP name is of the form:

> **USMC PKE PP with** <packages included in the PP, listed in the order in which they appear in the PP> **at EAL** <3 or 4, depending on the EAL selected> **with augmentation**

The words in bold print are included in every title, appropriate package names are listed for all of the packages included in the PP, and the EAL chosen is specified. Note that the list of packages in the title must be in the order in which they appear in this document in order to ensure consistency of naming. Also, when specifying a PP, only one PP from this family should be specified, i.e., the PP with the largest number of packages. The PP author should not attempt to specify all of the possible PPs represented in a single PP (which would include every possible combination of packages in the document). Instead, the PP author should name only the most comprehensive PP represented by the document.

The base functional requirements are defined in Section 5.1 and environmental requirements are defined in Section 5.2; appropriate assumptions, threats, and objectives are defined for the base set of requirements and environmental requirements in Sections 3 and 4. The base requirements, environmental requirements and associated assumptions, threats, and objectives must be included in every ST compliant with this PP.

The functional requirements for the packages are defined in Section 5.3 and corresponding assumptions, threats, and objectives are defined for each package in Sections 3 and 4. Each package represents a discrete set of threats, objectives, and requirements. The packages are named and their corresponding threats, objectives and functional requirements are identified in separate subsections within Sections 3.3, 4.3, and 5.3. When a package is included in an ST, all of the components of the package must be included, i.e., all of the threats, objective, requirements, and rationale. The ST author is expected to maintain the modularity of the packages in the ST, since this will enhance the ability to evaluate a TOE in a modular fashion.

The packages define a subset of X.509 certificate and revocation processing capabilities as defined in the ISO and Internet Engineering Task Force (IETF) standards. Some of the examples of these various capabilities include:

- Ability to process certificatePolicies extension
- Ability to process all certificate policies related extensions

- Ability to process name constraints extension
- Ability to handle the various public key algorithms (e.g., Rivest, Shamir, Adelman (RSA); Digital Signature Algorithm (DSA); Diffie Hellman (DH); Elliptic Curve Diffie Hellman (ECDH); etc.)
- Ability to handle a variety of public key based mechanisms (e.g., signature generation, signature verification, encryption, decryption, entity authentication, etc.)

The packages provide the granularity for the above listed capabilities. The ST author is further provided a high degree of flexibility by the use of selections and assignments for the various security functional requirements.

2.2.2 Part 2 and Part 2 Extended Security Functional Requirements

Using Part 2 of the CC as the tool for specifying security relevant requirements, this family of PPs addresses only the security aspects of PK enablement. For example, the PP does not deal with mechanisms of how the certificates and Certificate Revocation Lists (CRLs) are obtained since the security of certificates and CRLs does not depend on where or how they were obtained; their security is ensured through verification of digital signatures.

In the area of certification path validation, requirements are defined that are compliant with both the ISO X.509 and IETF PKIX Request for Comment (RFC) 2459. However, the certification path validation in these standards is procedural. In order to make the PP implementation neutral, certification path validation requirements are specified using non-procedural techniques.

CC access control related components are not appropriate to express the certificate and revocation information (e.g., Certificate Revocation List (CRL), OCSP response, etc.) processing requirements and hence CC Part 2 was extended to address the processing of certificates and revocation information.

2.2.3 Technical Approach for PKI requirements

This subsection describes the technical approached used in selecting and developing the PKI requirements.

The certification path validation requirements were developed with meaningful names for the components to define X.509 input, processing, and output segments. Certificate policy calculation is included in the output components.

An analysis of X.509 certificate processing revealed that a set of processing rules are applied to all the certificates and some additional rules are applied to intermediate (i.e., CA) certificates. Thus, basic certificate processing and intermediate certificate processing components have been established.

Neither X.509 nor PKIX require any trust anchor processing rules. However, to provide a tool that can be used to specify rules for trust anchor processing, trust anchor processing rules (including “none”) may be defined by the ST author as a part of the path validation initialisation.

The cryptographic operations that require the use of a public key must use the public key, public key parameters (if applicable) and subscriber identifying information from

certification path validation in order to preserve the security. Functional packages for the various cryptographic operations have been developed to specify this linkage.

This PP family provides functional requirements for processing all of the certificate extensions and for complete certification path validation. These capabilities are not required for every TOE, since this functionality accommodates the future plans for the DoD to cross-certify with other domains. This cross-certification will require use of the full X.509 extension set, including policyMapping, policyConstraints, and nameConstraints. While this PP family provides the ability to evaluate PKE applications (TOEs) that perform full X.509 path validation, it also provides the flexibility to evaluate applications (TOEs) that perform minimal to no policy and other extensions processing.

This family of PPs provides the capability to select public key cryptography algorithms since in the future the DoD may use a variety of algorithms. Packages for the public key cryptography algorithms are provided so that this family of PPs need not be revised to accommodate new algorithms.

The scope of this family of PPs excludes key recovery infrastructure-related functions since key recovery is an infrastructure function as opposed to a PKI application function. The ability to deal with multiple keys using the key identifier is addressed in appropriate locations in certificate path validation output and in cryptographic operations. The PKE application could have multiple keys due a variety of reasons such as key recovery, key history and re-key.

This PP family does not require a trusted or evaluated platform for PKE application execution. Rather, the approach is to specify:

- The self-protection and isolation requirements in the base requirements.
- The identification and authentication requirements in the base requirements.
- The access control requirements in the base requirements.
- The audit requirements (as required) in the Audit Management Package.
- The residual information protection for private and secret keys, which will be satisfied by a FIPS validated cryptographic module since the FIPS validated cryptographic module must provide for plaintext private secret keys to be zeroized.

It should be noted that some requirements, e.g., the Audit Management requirements, may be met by the environment, which might include a trusted operating system and/or FIPS 140 series validated cryptographic module.

The following features are deferred for future revisions of this family of PPs:

- Processing partitioned CRLs
- Processing delta CRLs
- Processing indirect CRLs
- Processing server based validation responses, such as Simple Certificate Validation Protocol (SCVP), OCSP Version 2, etc.

2.2.4 Specifying and Evaluating a PP or Compliant ST from this PP Family

When several PPs can be constructed using some or any combination of component packages, it is desirable to minimize the number of evaluations, e.g., in the case of this PP family, thousands of evaluations would be required to evaluate separately every possible PP that can be specified. To illustrate, if there are n packages, there are $2^n - 1$ PPs. Clearly, even for a small number of packages, it becomes a very large number of possible PPs. In naming a PP or specifying compliance with a PP, the author must use the naming convention defined in the Foreword and repeated in Section 2.2.1. In particular, packages listed in the title must be specified in the order in which they occur in the PP and only one PP from this family, the most comprehensive PP, should be specified, i.e., the PP with the largest number of packages. The PP author should not attempt to specify compliance with all of the possible PPs in the PP family to which compliance might be claimed, instead, compliance should be claimed only for the most comprehensive PP.

When claiming compliance with a particular PP, it is sufficient for an ST to identify any PP in this PP family by simply naming the PP. This is sufficient because the name of the PP clearly identifies all of the packages contained in the PP and the EAL. The ST evaluator can then evaluate compliance with the PP by examining the ST and its compliance with the PP packages and EAL identified in the title.

The approach used for this family of PPs is, during the PP family evaluation, to evaluate each package once, to evaluate inter-relationships among all packages once, and then to be confident about the validity of any PPs derived from this PP family. A PP derived from this family is considered to have passed the evaluation without any further work because in this PP family:

  - The packages are constructed with constraints as described below under Section 2.2.4.1, Constraints,
  - Each package is evaluated per the CEM; and
  - The packages go through the additional evaluation work units during PP family evaluation described below under Section 2.2.4.3, Additional Evaluation Work Units.
- A unique name is generated for the PP using the algorithm described above.
- An ETR is produced during the PP family evaluation that is valid for all PPs derived from this family because the ETR covers all of the packages. Note that in the case of this PP family, two ETRs may be required: one for each EAL.

2.2.4.1 Constraints

The following constraints were met in the development of this PP family:

1. Each package is complete, i.e., each package contains a name, TOE Description, threats, organization security policy (if applicable), secure usage assumptions (if applicable), security objectives for the TOE (if applicable), security objectives for the environment (if applicable), security functional requirements for the TOE (if applicable), IT security functional requirements for the environment (if applicable), non-IT security functional requirements for the

environment (if applicable), security assurance requirements, security objectives rationale, security requirements rationale, dependencies rationale, and strength of function rationale. In other words, the package has all of the components of a PP.

2. A dependency rationale points to other packages to satisfy some of the requirements. Note that dependencies are specifically identified for packages both in Section 2.3 and in Section 5.3 of this document. Also, the requirement that dependencies must be included is repeated in Section 2.3 and in Section 5.3.
3. Some material is included in a package by reference. For example, if assurance requirements and strength of function requirements are common to some or all packages, it is sufficient to include them only once as long as it is clear which packages are applicable.
4. From the TOE description, it is obvious that the security functionality provided by each package is different from functionality provided by other packages under evaluation.
5. The threats for each package are different from the threats for other packages. This means:
    a. A threat name appears in only one package, and
    b. Each threat description is distinct.
6. The objectives for each package are different from the objectives for other packages. This means:
    a. An objective name appears in only one package, and
    b. Each objective description is distinct.
7. The security functional requirements and security assurance requirements for all of the packages have the same label if and only if they are identical.
8. The authors describe the algorithm for naming the various composite PPs and show that they result in unique name for each possible composite PP.

2.2.4.2 Evaluating this PP family

In order for evaluate this family of PPs, the evaluator must do the following:

- The evaluator must evaluate the packages to verify that the assertions made in the previous section hold true.
- The evaluator must ensure that combining the packages will continue to be safe.
- The evaluator must verify that all the constraints listed above are satisfied by the packages.

A high-level methodology to perform this evaluation is described below.

For constraint items 1, 2, and 3 listed above, validation of these items falls naturally out of the evaluation of each package, as if that package or component were being

evaluated in a normal PP evaluation. Thus, if each package passes the evaluation, items 1, 2, and 3 are satisfied.

For constraint item 4, the evaluator should compare the functions performed by the various packages. The functions must be distinct. The functions may be distinct in terms of one or more of the following:

- Security capability; or
- Security services; or
- Data to which the security capability and/or service applies.

Constraint items 5, 6, and 7 can be executed using current CEM work units by treating the packages as if they are combined into a single composite PP. By analyzing all of the threats, objectives, and requirements at once, as if they were all contained in a single PP, any interactions or overlap between them can be identified.

For constraint item 8, the evaluator shall examine the composite PP and verify that the composite PP naming scheme will provide unique and unambiguous names. To perform this work unit, the evaluator will analyze the algorithm to make sure that the name clearly implies the packages that are either included, excluded or both. The evaluator shall also take some sample cases and see that each case results in a unique, meaningful and unambiguous name.

2.2.4.3 Additional Evaluation Work Units for this PP Family

The following additional work units must be carried out to ensure that when the packages are combined, the evaluation will continue to be valid.

1. The evaluator shall verify that the security objectives for the TOE and security objectives for the Environment do not conflict. The evaluator shall look at all the objectives for the packages and/or components collectively and apply the methodology used for APE_OBJ.1-9 to ensure that the objectives do not conflict.
2. The evaluator shall verify that the IT security requirements do not conflict. The evaluator shall look at all the IT security requirements for the packages and/or components collectively and apply the methodology used for APE_REQ.1-22 to ensure that the IT security requirements do not conflict.
3. If the same requirement appears in more than one package, it applies to mutually exclusive scope, e.g., to different data.
4. The evaluator shall examine the packages to ensure that either the iterations of the same component are properly applied or there is sufficiently detailed guidance provided to the ST author in order to uniquely and unambiguously label each iterated component.

2.3 Definition of TOE

For all of the PPs in this PP family, the TOE is defined as the PKE application. The TOE and TSF boundaries will be defined by the ST author and will address what functionality is included in the TOE and what is included in the environment. The PKE application (TOE) must include, either as part of the environment or as part of the application itself,

access control functionality, identification and authentication, and security management functions, which ensure the security of the application and its data.

All of the PPs in this family assume that the TOE environment includes one or more cryptographic module(s) that are all validated at FIPS 140 series Level 1 or greater. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, Hash based Message Authentication Code (HMAC) and/or other required cryptographic functions. Note that the TOE environment may contain more that one cryptographic module so that some functions, such as key pair generation, may be performed in a hardware cryptographic module, while others, such as secure hash, may be performed in a software module. Generally, private key operations will be performed in the hardware cryptographic module and public key and symmetric key operations will be performed either in the hardware or the software cryptographic module.

This PP family also assumes that certificates and status message, i.e., CRLs or OCSP responses are available as part of the DoD PKI interface.

For all of the PPs in this PP family, TOE user data is defined as any data that is encrypted, decrypted, signed, verified, imported or exported by the user. TOE user data may also include the user's cryptographic keys. The ST author will provide a specific definition of user data, depending upon the application (TOE).

For all of the PPs in this PP family, TSF data is defined as identification and authentication data, private keys owned by the system, security attributes and other data as defined by the ST author. Note that if the environment performs the identification and authentication function or other security functions, then the associated data is not considered to be TSF data, since it is not within the TOE boundary.

This PP family defines a set of security requirements to be levied on TOEs. A TOE may be a stand-alone system or consist of components in a network or a distributed environment. The TOE may consist of an application running on one or more processors and associated peripherals and storage devices to be used by multiple users to perform a variety of PKI functions requiring controlled, shared access to the information stored on the system. The ST author will provide a specific definition of the TOE.

All of the PPs in this PP family contain a set of "base" functionality. The base functionality specifies 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 PKCS-7 type information generation. The base functionality also maintains secure storage of trust anchors. It should be noted that some or all of the base functionality may be provided by the environment, e.g., a trusted operating system and/or FIPS 140 series validated cryptographic module.

As stated above, all of the PPs in this PP family include the base functionality. Assumptions, threats, objectives, and requirements are defined in the following sections for the base functionality.

Table 2.1 provides a summary of the functionality contained in the packages included in this PP family. The following subsections describe the functionality of the packages. Note that each of the packages described in the following subsections have an assurance level of either EAL 3 augmented or EAL 4 augmented.

Note that some packages have dependencies on other packages, i.e., when a package with dependencies is included in a PP, the dependent package(s) must also be included in their entirety. A valid PP must contain all dependencies defined for packages in the PP. A summary of package dependencies is as follows:

- Certification Path Validation – Basic Package is a dependency of the following other packages, i.e., when the following packages are included in a PP, the Certification Path Validation – Basic Package must also be included in the PP:
  - Certification Path Validation – Basic Policy Package
  - Certification Path Validation – Policy Mapping Package
  - Certification Path Validation – Name Constraints Package
  - PKI Encryption using Key Transfer Algorithms
  - PKI Encryption using Key Agreement Algorithms
  - PKI Decryption using Key Agreement Algorithms
  - PKI Signature Verification
  - PKI Based Entity Authentication
  - Continuous Authentication
- Certification Path Validation – Basic Policy is a dependency of Certification Path Validation – Policy Mapping
- PKI Based Entity Authentication package is a dependency of Continuous Authentication Package

Table 2.1 lists any dependent packages for each of the packages included in this PP family. Note that if a package with dependencies is included in a PP or ST, then the dependency package(s) must also be included in the PP.

Table 2.1 – Summary of Packages

Package Name	Functionality	Dependency
Certification Path Validation (CPV) – Basic	Perform all X.509 validation checks except policy processing and name constraints processing	None
CPV – Basic Policy	Process certificatePolicies extension	CPV – Basic
CPV – Policy Mapping	Process policy mapping related extensions: policyMapping, policyConstraints, and inhibitAnyPolicy	CPV – Basic, CPV – Basic Policy
CPV – Name Constraints	Process nameConstraints extension	CPV – Basic

Package Name	Functionality	Dependency
Constraints		
PKI Signature Generation	Use private key for signature generation Generate the signature information (e.g., Public Key Cryptography Standard 7 (PKCS 7) blob)	None
PKI Signature Verification	Process the signature information (e.g., PKCS 7 blob) Use public key to verify signature	CPV – Basic
PKI Encryption using Key Transfer Algorithms	Generate the encryption envelope information (e.g., PKCS 7 blob) Use public key for encryption	CPV – Basic
PKI Encryption using Key Agreement Algorithms	Generate the key agreement envelope information (e.g., PKCS 7 blob) Use decryptor public key for key agreement Use encryptor private key for key agreement	CPV – Basic
PKI Decryption using Key Transfer Algorithms	Process encryption envelope information (e.g., PKCS 7 blob) Use private key for decryption	None
PKI Decryption using Key Agreement Algorithms	Process the key agreement envelope information (e.g., PKCS 7 blob) Use encryptor public key for key agreement Use decryptor private key for key agreement	CPV – Basic
PKI Based Entity Authentication	Carry out the “assigned” authentication protocol(s) Use public key for authentication	CPV – Basic
Online Certificate Status Protocol Client	Generate OCSP request in accordance with RFC 2560 Process OCSP response	None
Certificate Revocation List (CRL) Validation	Obtain CRL Process CRL	None
Audit Management	Generate Audit Log Protect Audit Log Generate human readable audit reports	None
Continuous Authentication	Perform Continuous Authentication	PKI Based Entity Authentication, CPV - Basic

2.3.1 Certification Path Validation – Basic Package

The Certification Path Validation – Basic Package (CPV – Basic) provides for all X.509 validation checks except policy processing and name constraints processing. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented. This package addresses the validation of the certification path and certification path development. The most likely implementation consists of developing a path (using a variety of techniques) and then validating the certification path. Certification path validation generally consists of validating certificates starting with the one certified by a trust anchor and ending with the one issued to the subscriber of interest. However, in order to be implementation neutral, this package does not mandate any ordering of certification path development and certification validation processes. A compliant implementation will only need to meet the security requirements specified in this package.

All processing defined is X.509 and PKIX compliant.

There are three types of public key certificates:

- Trust anchors: These are self-signed certificates that do not require any validation. The trust anchor (self-signed certificate) is generally in the form of a certificate. The primary purpose of the trust anchor is to obtain the Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable). This package permits validation of trust anchor, including validating signature and verifying that the trust anchor 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 includes processes for the following security related certificate extensions checks: no-check, keyUsage, extendedKeyUsage, and basicConstraints.

This version of the PP family assumes that the path validation is being done as of current time (as opposed to, e.g., verification of old signature in case of dispute). Future versions may include the capability to validate path as of a user-defined time.

2.3.2 Certification Path Validation – Basic Policy Package

The Certification Path Validation – Basic Policy package is dependent on the CPV – Basic package. The functionality in this package is the processing of certificatePolicies extension. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.3 Certification Path Validation – Policy Mapping Package

The Certification Path Validation – Policy Mapping package is dependent on the CPV – Basic Policy and the CPV – Basic packages. The functionality in this package is the processing of the following certificate policies related extension: policyMapping,

inhibitAnyPolicy, and policyConstraints. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.4 Certification Path Validation – Name Constraints Package

The Certification Path Validation – Name Constraints is dependent on the CPV – Basic package. The functionality in this package is the processing of nameConstraints extension. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.5 PKI Signature Generation Package

The PKI Signature Generation package provides functionality to use the private key for signature generation and to generate the signature information. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.6 PKI Signature Verification Package

The PKI Signature Verification package is dependent on the CPV – Basic package. This package provides functionality for processing the signature information, e.g. the PKCS 7 blob, and using the public key to verify a signature. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.7 PKI Encryption using Key Transfer Algorithms Package

The PKI Encryption using Key Transfer Algorithms package is dependent on the CPV – Basic package. The package provides functionality for performing public key encryption using key transfer algorithms such as RSA. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.8 PKI Encryption using Key Agreement Algorithms Package

The PKI Encryption using Key Agreement Algorithms package is dependent on the CPV – Basic package. This package provides functionality to perform key encryption using key agreement algorithms such as DH or ECDH. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.9 PKI Decryption using Key Transfer Algorithms Package

The PKI Decryption using Key Transfer Algorithms package provides functionality to perform public key decryption using key transfer algorithms such as RSA. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented. Since only the decrypting party's private key is used, this package does not require certificate path processing functionality.

2.3.10 PKI Decryption using Key Agreement Algorithms Package

The PKI Decryption using Key Agreement Algorithms package is dependent on the CPV – Basic package. This package provides the functionality to perform key decryption

using key agreement algorithms such as DH or ECDH. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.11 PKI Based Entity Authentication Package

The PKI Based Entity Authentication is dependent on the CPV – Basic package and allows PKI to be used for an entity authentication service. This package allows the ST author to select a PKI based entity authentication standard for identification and authentication of a remote entity. This package shall be used for initial authentication of the entity. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.12 Online Certificate Status Protocol Client Package

The Online Certificate Status Protocol Client package allows the TOE to make Online Certificate Status Protocol (OSCP) requests and to validate 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. The ST author can assign additional rules to process OCSP extensions. If the OCSP implementation establishes trust in the OCSP responder by performing Certificate Path Validation, then the CPV – Basic package may be used in combination with this package. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.13 Certificate Revocation List (CRL) Validation Package

The Certificate Revocation List (CRL) Validation package allows the TOE to validate a CRL. This version of this package does not require processing of a CRL issuing distribution point (IDP) CRL or a delta CRL. Future versions may include that capability by codifying Annex B of X.509 standard.

It should be noted that this package may be used to process a CRL that is pointed to by a CRL Distribution Point (CRLDP) extension in a certificate as long as the CRL is a full CRL, indicated by the absence of IDP and deltaCRLIndicator extensions.

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. If the compliant implementation develops a certification path, then the CPV – Basic package may be used in combination with this package. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.14 Audit Management Package

The Audit Management package generates and protects audit events relevant to the TOE. Examples of audit events are:

- Management of trust anchors (addition, deletion)
- Identification and Authentication

- Signature verification success, date and time, and policies under which signatures were valid
- Signature verification failure, date and time, cause of failure (signature on the object failed, certification path failure, policy failure, etc.)
- User override events (current CRL availability, accept policy failure, accept null policy, accept other policy, etc.)

The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.3.15 Continuous Authentication Package

This package is dependent on PKI Based Entity Authentication and the CPV – Basic packages. This package is used for continuous authentication of the protocol, command, packets etc. The functionality in this package is the same regardless of whether the assurance level is EAL 3 augmented or EAL 4 augmented.

2.4 Assurance Requirements

There are two assurance levels included in this family of PPs: the first is EAL 3 with augmentation, and the second is EAL 4 with augmentation. The ST author will determine the appropriate assurance requirements, based on application requirements. The EAL 3 with augmentation PPs will be selected when the TOE requires a moderate level of independently assured security and requires a thorough investigation of the TOE and its development without substantial re-engineering. The EAL 4 with augmentation PPs will be selected in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs.

3 TOE Security Environment

3.1 Secure Usage Assumptions for all PPs in this PP family

Table 3.1 lists the Secure Usage Assumptions for the IT environment. These assumptions for the IT environment are included in every PP in this PP family.

Table 3.1 – Assumptions for the IT Environment

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

3.2 Base Threats to Security for all PPs in this PP Family

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

The following threats are included in every PP in this PP family. These threats must be included in every ST that claims compliance any one of the PPs in this family.

Table 3.2 – Base Threats to Security for all PPs in this PP Family

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

3.3 Threats to Security for Packages

The following subsections define 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, or 2) failure of the TOE.

Note that in addition to the threats defined below for each package, every PP derived from this PP family also includes the base threats defined in Table 3.2.

3.3.1 Certification Path Validation – Basic Package

In addition to the base threats, the following threats are defined for the Certification Path Validation – Basic package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.3 – Threats for the CPV – Basic Package

#	Threat Name	Threat Description
1	T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.
2	T.DOS_CPV_Basic	The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.
3	T.Expired_Certificate	An expired (and possibly revoked) certificate could be used for signature verification.
4	T.Masquarade	An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.
5	T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.
6	T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.
7	T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.
8	T.User_CA	A user could act as a CA, issuing unauthorized certificates.

3.3.2 Certification Path Validation – Basic Policy Package

The following threats are defined for the Certification Path Validation – Basic Policy package. This threat applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.4 – Threats for the CPV – Basic Policy Package

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

3.3.3 Certification Path Validation – Policy Mapping Package

The following threats are defined for the Certification Path Validation – Policy Mapping package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.5 – Threats for the CPV – Policy Mapping Package

#	Threat Name	Threat Description
1	T.Mapping	The user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping.
2	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.

3.3.4 Certification Path Validation – Name Constraints Package

The following threats are defined for the Certification Path Validation – Name Constraints Package. This threat applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.6 – Threats for the CPV – Name Constraints Package

#	Threat Name	Threat Description
1	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.

3.3.5 PKI Signature Generation Package

The following threats are defined for the PKI Signature Generation package. This threat applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.7 – Threats for the PKI Signature Generation Package

#	Threat Name	Threat Description
1	T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature in absence of hint.

3.3.6 PKI Signature Verification Package

The following threats are defined for the PKI Signature Verification Package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.8 – Threats for the PKI Signature Verification Package

#	Threat Name	Threat Description
1	T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.
2	T.Clueless_PKI_Ver	The user may try only inappropriate certificates for verification in absence of hint.

3.3.7 PKI Encryption using Key Transfer Algorithms Package

The following threats are defined for the PKI Encryption using Key Transfer Algorithms Package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

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

#	Threat Name	Threat Description
1	T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.
2	T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.

3.3.8 PKI Encryption using Key Agreement Algorithms Package

The following threats are defined for the PKI Encryption using Key Agreement Algorithms package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.10 – Threats for the PKI Encryption using Key Agreement Algorithms Package

#	Threat Name	Threat Description
1	T.Assumed_Identity_With_En	A user may assume the identity of another user in order to perform encryption using Key Agreement algorithms.
2	T.Clueless_With_En	The user may try only inappropriate certificates for encryption using Key Agreement algorithms in absence of hint.

3.3.9 PKI Decryption using Key Transfer Algorithms Package

The following threats are defined for the PKI Decryption using Key Transfer Algorithms package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.11 – Threats for the PKI Decryption using Key Transfer Algorithms Package

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

3.3.10 PKI Decryption using Key Agreement Algorithms Package

The following threats are defined for the PKI Decryption using Key Agreement Algorithms package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.12 – Threats for the PKI Decryption using Key Agreement Algorithms Package

#	Threat Name	Threat Description
1	T.Assumed_Identity_With_De	A user may assume the identity of another user for decrypting using Key Agreement algorithms.
2	T.Clueless_With_De	The user may try only inappropriate certificates for decryption using Key Agreement algorithms in absence of hint.
3	T.Garble_With_De	The user may not apply the correct key agreement algorithm or private key, resulting in garbled data.

3.3.11 PKI Based Entity Authentication Package

The following threats are defined for the PKI Based Entity Authentication package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.13 – Threats for the PKI Based Entity Authentication Package

#	Threat Name	Threat Description
1	T.Assumed_Identity_Auth	A user may assume the identity of another user to perform entity based authentication.
2	T.Replay_Entity	An unauthorized user may replay valid entity authentication data.

3.3.12 Online Certificate Status Protocol Client Package

The following threats are defined for Online Certificate Status Protocol Client package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.14 – Threats for the OCSP Client Package

#	Threat Name	Threat Description
1	T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.
2	T.Replay_OCSP_Info	The user may accept an old OCSP response resulting in accepting a currently revoked certificate.
3	T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.

3.3.13 Certificate Revocation List (CRL) Validation Package

The following threats are defined for the Certificate Revocation List (CRL) Validation package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.15 – Threats for the Certificate Revocation List (CRL) Validation Package

#	Threat Name	Threat Description
1	T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.

#	Threat Name	Threat Description
2	T.Replay_Revoc_Info_CRL	The user may accept an old CRL resulting in accepting a currently revoked certificate.
3	T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL.

3.3.14 Audit Management Package

The following threats are defined for the Audit Management package. These threats apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.16 – Threats for the Audit Management Package

#	Threat Name	Threat Description
1	T.Accountability	The security relevant audit events cannot be linked to individual actions.
2	T.Audit_Excess	The security audit log has excessive data for analysis.
3	T.Audit_Fill	The security audit log gets filled too fast to be of practical use.
4	T.Audit_Modify	The accuracy of the security audit log cannot be trusted since unauthorized modification may have been made.
5	T.Audit_Unreadable	The audit log cannot be read and interpreted by human beings and hence security relevant events cannot be investigated.
6	T.No_Audit	There is no audit log to investigate security relevant events.

3.3.15 Continuous Authentication Package

The following threat is defined for Continuous Authentication package. This threat applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 3.17 – Threats for the Continuous Authentication Package

#	Threat Name	Threat Description
1	T.Hijack	An unauthorized user may hijack an authenticated session.

4 Security Objectives

4.1 Base Security Objectives for the TOE

Base TOE security objectives are defined in Table 4.1, below. The base TOE security objectives are included in every PP in this PP family and must be included in every ST that claims compliance with any PP in this family. Note that base TOE security objectives may be met by the environment and, in that case, should be stated in the ST as "OE" prefixed objectives as opposed to "O" prefixed objectives.

Table 4.1 – Security Objectives for the TOE for all PPs in this PP Family

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

4.2 Security Objectives for the Environment

Table 4.2 lists security objectives for the environment. These environmental objectives are included in every PP in this PP family and must be included in any ST that claims compliance to this family of PPs.

Table 4.2 – Security Objectives for the Environment

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

4.3 Security Objectives for Packages

Security objectives for the packages in this PP family are defined in the following subsections. Note that in addition to the security objectives defined for each individual package, each PP derived from this PP family must include the base security objectives for the TOE defined in Section 4.1 and the environmental objectives defined in Section 4.2.

4.3.1 Certification Path Validation – Basic Package

The following security objectives are defined for the Certification Path Validation – Basic PPs. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.3 – Security Objectives for CPV – Basic Package

#	Objective Name	Objective Description
1	O.Availability	The TSF shall continue to provide security services even if revocation information is not available.
2	O.Correct_Time	The TSF shall provide accurate temporal validation results.
3	O.Current_Certificate	The TSF shall only accept certificates that are not expired.
4	O.Get_KeyInfo	The TSF shall provide the user public key and related information in order to carry out cryptographic functions.
5	O.Path_Find	The TSF shall be able to find a certification path from a trust anchor to the subscriber.
6	O.Trusted_Keys	The TSF shall use trusted public keys in certification path validation.
7	O.User	The TSF shall only accept certificates issued by a CA.
8	O.Verified_Certificate	The TSF shall only accept certificates with verifiable signatures.
9	O.Valid_Certificate	The TSF shall use certificates that are valid, i.e., not revoked.

4.3.2 Certification Path Validation – Basic Policy Package

The following security objective is defined for the Certification Path Validation – Basic Policy package. This security objective applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.4 – Security Objectives for CPV – Basic Policy Package

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

4.3.3 Certification Path Validation – Policy Mapping Package

The following security objectives are defined for the Certification Path Validation – Policy Mapping package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.5 – Security Objectives for CPV – Policy Mapping Package

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

4.3.4 Certification Path Validation – Name Constraints Package

The following security objective is defined for the Certification Path Validation – Name Constraints package. This security objective applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.6 – Security Objectives for CPV – Name Constraints Package

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

4.3.5 PKI Signature Generation Package

The following security objective is defined for the PKI Signature Generation package. This security objective applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.7 – Security Objectives for PKI Signature Generation Package

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

4.3.6 PKI Signature Verification Package

The following security objectives are defined for the PKI Signature Verification package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.8 – Security Objectives for PKI Signature Verification Package

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

4.3.7 PKI Encryption using Key Transfer Algorithms Package

The following security objectives are defined for the PKI Encryption using Key Transfer Algorithms package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

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

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

4.3.8 PKI Encryption using Key Agreement Algorithms Package

The following security objectives are defined for the PKI Encryption using Key Agreement Algorithms package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.10 – Security Objectives for PKI Encryption using Key Agreement Algorithms Package

#	Objective Name	Objective Description
1	O.Hints_Enc_W	The TSF shall provide hints for selecting correct certificates or keys for PKI encryption using Key Agreement algorithms.
2	O.Linkage_Enc_W	The TSF shall use the correct user public key for key agreement during encryption.

4.3.9 PKI Decryption using Key Transfer Algorithms Package

The following security objectives are defined for the PKI Decryption using Key Transfer Algorithms package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.11 – Security Objectives for PKI Decryption using Key Transfer Algorithms Package

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

4.3.10 PKI Decryption using Key Agreement Algorithms Package

The following security objectives are defined for the PKI Decryption using Key Agreement Algorithms package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.12 – Security Objectives for PKI Decryption using Key Agreement Algorithms Package

#	Objective Name	Objective Description
1	O.Hints_Dec_W	The TSF shall provide hints for selecting correct certificates or keys for PKI decryption using Key Agreement algorithms.
2	O.Linkage_Dec_W	The TSF shall use the correct user public key for key agreement during decryption.
3	O.Correct_KA	The TSF shall use appropriate private key and key agreement algorithm.

4.3.11 PKI Based Entity Authentication Package

The following security objectives are defined for the PKI Based Entity Authentication package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.13 – Security Objectives for PKI Based Entity Authentication Package

#	Objective Name	Objective Description
1	O.I&A_Remote	The TSF shall uniquely identify all remote entities, and shall authenticate the claimed identify before granting a remote entity access to the TOE facilities.
2	O.Limit_Actions_Auth_Remote	The TSF shall restrict the actions a remote entity may perform before the TSF verifies the identity of the remote entity.
3	O.Linkage	The TSF shall use the correct user public key for authentication.
4	O.Single_Use_I&A	The TSF shall use the I&A mechanism that requires unique authentication information for each I&A.

4.3.12 Online Certificate Status Protocol Client Package

The following security objectives are defined for the Online Certificate Status Protocol Client package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.14 – Security Objectives for Online Certificate Status Protocol Client Package

#	Objective Name	Objective Description
1	O.Accurate_OCSP_Info	The TSF shall accept only accurate OCSP responses.
2	O.Auth_OCSP_Info	The TSF shall accept the revocation information from an authorized source for OCSP transactions.
3	O.Fresh_OCSP_Info	The TSF accept only reasonably current revocation information for OCSP transactions.
4	O.User_Override_Fresh_OCSP	The TSF shall permit the user to override the freshness requirement for the OCSP response.

4.3.13 Certificate Revocation List (CRL) Validation Package

The following security objectives are defined for the Certificate Revocation List Validation Package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.15 – Security Objectives for Certificate Revocation List (CRL) Validation Package

#	Objective Name	Objective Description
1	O.Accurate_Rev_Info	The TSF shall accept only accurate revocation information.
2	O.Auth_Rev_Info	The TSF shall accept the revocation information from an authorized source for CRL.
3	O.Fresh_Rev_Info	The TSF shall accept only reasonably current CRL .
4	O.User_Override_Fresh_CRL	The TSF shall permit the user to override the freshness requirement for CRL.

4.3.14 Audit Management Package

The following security objectives are defined for the Audit Management Package. These security objectives apply to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.16 – Security Objectives for Audit Management Package

#	Objective Name	Objective Description
1	O.Audit	The TSF shall audit security relevant events.
2	O.Audit_Protect	The TSF shall protect the security audit log from unauthorized modifications.
3	O.Audit_Readable	The TSF shall be able to generate human readable reports from the audit log.
4	O.Audit_Select	The TSF shall permit authorized users to select auditable events.
5	O.Audit_User	The TSF shall be capable of associating audit events with individual users.

4.3.15 Continuous Authentication Package

The following security objective is defined for the Continuous Authentication package. This security objective applies to this package at both EAL 3 Augmented and EAL 4 augmented assurance levels.

Table 4.17 – Security Objectives for Continuous Authentication Package

#	Objective Name	Objective Description
1	O.Continuous_I&A	The TSF shall continuously authenticate the entity.

5 IT Security Requirements

This section defines the TOE security functional requirements and assurance requirements, included for all of the PPs in this PP family. Requirements are drawn from the CC Parts 2 and 3 and have been written as required as Part 2 extended requirements. Selections and assignments to be made by the ST author in Part 2 and Part 2 extended requirements are enclosed in [square brackets] and text is in *italics*. A list of selections, identified as "Selection by the ST author," allow the ST author to select one or more of the items listed as indicated. Assignments, identified as "Assignment by the ST author," provide the ST author with the opportunity to insert specific information. Where the PP authors have made refinements in Part 2 requirements, the text is indicated by ***bold italics***. Assignments and selections in Part 2 requirements are indicated by *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.

Each PP in this family of PPs is Part 2 extended. The definition of Part 2 extended is found in the CC Part 3, section 5.4, "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 the family of PPs are listed in Table 5.1, below. Part 2 extended requirements are explicitly identified as "Part 2 extended." Each PP in this family of PPs uses security functional requirements from the table below. In other words, each PP in this family of PPs uses a subset of security functional requirements from the table below.

Table 5.1 – Part 2 or Part 2 Extended Requirements

Requirement	Part 2 or extended
FAU_GEN.1	Part 2
FAU_GEN.2	Part 2
FAU_SAR.1	Part 2
FAU_SEL.1	Part 2
FAU_STG.1	Part 2
FDP_ACC.1	Part 2
FDP_ACF.1	Part 2
FIA_AFL.1	Part 2
FIA_ATD.1	Part 2
FIA_UAU.1	Part 2
FIA_UAU.4	Part 2
FIA_UAU.6	Part 2
FIA_UAU.7	Part 2
FIA_UID.1	Part 2

Requirement	Part 2 or extended
FMT_MSA.1	Part 2
FMT_MSA.3	Part 2
FMT_MTD.1	Part 2
FMT_SMF.1	Part 2
FMT_SMR.2	Part 2
FPT_RVM.1	Part 2
FPT_SEP.1	Part 2
FPT_STM.1	Part 2
FCS_CRM_FPS.1	Part 2 Extended
FDP_CPD.1	Part 2 Extended
FDP_DAU_CPV_CER.1	Part 2 Extended
FDP_DAU_CPV_CER.2	Part 2 Extended
FDP_DAU_CPV_CER.3	Part 2 Extended
FDP_DAU_CPV_CER.4	Part 2 Extended
FDP_DAU_CPV_CER.5	Part 2 Extended
FDP_DAU_CPV_INI.1	Part 2 Extended
FDP_DAU_CPV_INI.2	Part 2 Extended
FDP_DAU_CPV_INI.3	Part 2 Extended
FDP_DAU_CPV_INI.4	Part 2 Extended
FDP_DAU_CPV_OUT.1	Part 2 Extended
FDP_DAU_CPV_OUT.2	Part 2 Extended
FDP_DAU_CPV_OUT.3	Part 2 Extended
FDP_DAU_CRL.1	Part 2 Extended
FDP_DAU_ENC.1	Part 2 Extended
FDP_DAU_ENC.2	Part 2 Extended
FDP_DAU_ENC.3	Part 2 Extended
FDP_DAU_OCS.1	Part 2 Extended
FDP_DAU_SIG.1	Part 2 Extended
FDP_ETC_ENC.1	Part 2 Extended
FDP_ETC_ENC.2	Part 2 Extended

Requirement	Part 2 or extended
FDP_ETC_SIG.1	Part 2 Extended
FDP_ITC_ENC.1	Part 2 Extended
FDP_ITC_ENC.2	Part 2 Extended
FDP_ITC_PKI_INF.1	Part 2 Extended
FDP_ITC_SIG.1	Part 2 Extended
FIA_UAU_SIG.1	Part 2 Extended

All of the PPs in this family contain a set of base security functional requirements and environmental requirements. These requirements, which are common to all of the PPs, are included in Section 5.1 and 5.2 below. There are 15 packages and 2 different EALs defined in this family of PPs. A PP in this family is composed of the following:

- Base requirements as defined in Section 5.1
- Environmental requirements defined in Section 5.2
- One or more of the fifteen PP functional requirements packages defined in Section 5.3
- Either EAL 3 augmented or EAL 4 augmented requirements defined in sections 5.4 and 5.5, respectively.

5.1 TOE Base Security Functional Requirements

A list of the base security functional requirements is provided in Table 5.2. The full text of the security functional requirements is contained below. The base requirements must be included in any PP in this PP family.

The base requirements 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 PKCS-7 type information generation. The base requirements also maintain secure storage of trust anchors.

It should be noted that some or all of the base requirements may be met by the environment such as a trusted operating system and/or FIPS 140 series validated cryptographic module.

Table 5.2 – TOE Base Security Functional Requirements included in all PPs in this PP Family

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

5.1.1 Class FDP – User Data Protection

FDP_ACC.1 Subset access control – PKI Credential Management

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the *PKI credential management SFP* on [assignment by the ST author: *list of subject*s, *objects, and operations among subjects and objects covered by the SFP*].

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

Dependencies: FDP_ACF.1 Security attribute based access control

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

Hierarchical to: No other components.

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

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed **[selection of one or more by the ST author:**

a) ***Private keys may be generated, imported, exported, destroyed, used by*** **[selection of one or more by the ST author*: owner, administrator,* [assignment by the ST author: *other roles defined by the ST author*]].**

b) ***Public key certificates may be imported, exported, deleted by [*selection of one or more by the ST author*: owner, administrator,* [assignment by the ST author: *other roles defined by the ST author*]].**

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

d) **[assignment by the ST author: *other rule(s)*].]**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment by the ST author: *rules, based on security attributes that explicitly authorize access of subjects to objec*ts].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [assignment by the ST author: *rules, based on security attributes that explicitly deny access of subjects to objec*ts].

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

5.1.2 Class FIA – Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components

FIA_AFL.1.1 The TSF shall detect when [assignment by the ST author: *number*] unsuccessful authentication attempts occur related to [assignment by the ST author: *list of authentication events*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment by the ST author: *list of actions*].

Dependencies: FIA_UAU.1 Timing of authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components

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

Dependencies: None

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components

FIA_UAU.1.1 The TSF shall allow [assignment by the ST author*: list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated.

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

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components

FIA_UAU.7.1 The TSF shall provide only [assignment by the ST author: *list of feedback*] to the user while the authentication is in progress.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UID.1 Timing of identification

Hierarchical to: No other components

FIA_UID.1.1 The TSF shall allow [assignment by the ST author*: list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.

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

Dependencies: None.

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

5.1.3 Class FMT – Security Management

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components

FMT_MSA.1.1 The TSF shall enforce the *PKI credential management SFP* to restrict the ability to [*selection of* ***one or more by the ST author****: change_default, query, modify, delete,* [assignment by the ST author*: other* ***specified*** *operations*]] the security attributes **[*selection of one or more by the ST author: user role, key identifier, association between private key and public key certificate,* [assignment by the ST author*: other security attributes*]]** to **[selection of one or more by the ST author*: owner, user, administrator,* [assignment by the ST author*: other role(s) defined*]]**.

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

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components

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

FMT_MSA.3.2 The TSF shall allow the **[selection of one or more by the ST author: *owner, user, administrator,* [assignment by the ST author: *other role(s) defined*]]** to specify alternative initial values to override the default values when an object or information is created.

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

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components

FMT_MTD.1.1 The TSF shall restrict the ability to [selection of **one or more by the ST author**: *change_default, modify, delete, clear,* ***import, add,*** [assignment by the ST author: *other operations*]] the **[selection of one or more by the ST author: *trust anchors, identification data, authentication data, number of unsuccessful authentication attempts* [assignment by the ST author: *other TSF data*]]** to **[selection of one or more by the ST author: *owner, user, administrator,* [assignment by the ST author: *other role(s) defined*]]**.

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

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

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [assignment by ST author*: list of security management functions to be provided by the TSF*].

Dependencies: None

FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1

FMT_SMR.2.1	The TSF shall maintain the roles **[selection of one or more by the ST author: *user, owner, administrator, remote entity* [assignment by the ST author: *other role(s) defined*]]**.
FMT_SMR.2.2	The TSF shall be able to associate users with roles.
FMT_SMR.2.3	The TSF shall ensure that the conditions [assignment by the ST author: *conditions for the different roles*] are satisfied.
Dependencies:	FIA_UID.1 Timing of identification

.

5.1.4 Class FPT – Protection of the TOE Security Functions

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components

FPT_RVM.1.1	The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the **TSF Scope of Control (TSC)** is allowed to proceed.
Dependencies:	None.

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components

FPT_SEP.1.1	The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1.2	The TSF shall enforce separation between the security domains of subjects in the TSC.
Dependencies:	None.

5.1.5 Strength of Function Requirement

The strength of function for the TOE authentication function is assumed to be SOF-basic. This SOF level matches an attack potential of low. The strength of cryptographic algorithms is outside the scope of the CC. Strength of function only applies to non-cryptographic, probabilistic or permutational mechanisms. The SOF requirement applies to the identification and authentication functionality within the TOE.

5.2 Security Functional Requirements for the IT Environment

The functions in this section address the security functional requirements for the IT environment. These requirements must be included in any PP in this PP family.

5.2.1 Class FCS – Cryptographic Support

FCS_CRM_FPS.1 FIPS compliant cryptographic module

Hierarchical to: No other components.

FCS_CRM_FPS.1.1	The IT environment shall provide all cryptographic modules necessary for the TSF.
FCS_CRM_FPS.1.2	Each cryptographic module shall be FIPS 140 series Level 1 validated.
Dependencies:	None.

5.2.2 Class FDP – User Data Protection

FDP_ITC_PKI_INF.1 Import of PKI information from outside the TSF

Hierarchical to: No other components.

FDP_ITC_PKI_INF.1.1	The IT environment shall ensure the availability of [selection of one or more by the ST author: certificates, CRLs, OCSP responses, [assignment by the ST author: *other PKI information*]], to the TOE [assignment: a defined availability metric] given the following conditions [selection of one or more by the ST author: *availability of network connection, availability of information server, availability of information in the application protocol, availability of information to the IT environment,* [assignment by the ST author: *other conditions to ensure availability*]].
Dependencies	None

5.2.3 Class FPT – Protection of the TSF

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1	The IT environment shall be able to provide reliable time stamps for TSF use.
Dependencies:	None.

5.3 Security Functional Requirements for Packages

The following subsections define functional requirements for each package. Note that all PPs in this PP family must include the base functional requirements defined in Section 5.1 and the environmental requirements defined in Section 5.2, in addition to the unique requirements defined below for the particular packages selected for inclusion. There are 14 subsections below. Each subsection provides functional requirements for a package. Note that some packages have dependencies on other packages. A summary of package dependencies is as follows:

- Certification Path Validation – Basic Package is a dependency of the following other packages, i.e., when the following packages are included, the Certification Path Validation – Basic Package must also be included:

- Certification Path Validation – Basic Policy Package
- Certification Path Validation – Policy Mapping Package
- Certification Path Validation – Name Constraints Package
- PKI Encryption using Key Transfer Algorithms
- PKI Encryption using Key Agreement Algorithms
- PKI Decryption using Key Agreement Algorithms
- PKI Signature Verification
- PKI Based Entity Authentication
- Continuous Authentication

- Certification Path Validation – Basic Policy is a dependency of Certification Path Validation – Policy Mapping Package
- PKI Based Entity Authentication is a dependency of Continuous Authentication Package

Note that functional requirements for packages remain the same, regardless of whether EAL 3 augmented or EAL 4 augmented is selected as the assurance level.

A summary of the functional requirements included in each package and package dependencies is provided in Table 5.3, below. Note that if a package has one or more dependency packages listed, then all the dependency package(s) must be included in the PP or ST when the dependent package is included in the PP. It is not valid under any circumstances to include a package with dependencies and not include the dependency packages in the PP or ST, i.e. dependencies must be included as specified in Table 5.3.

Table 5.3 – Summary of Security Functional Requirements in Packages

Package Name	Functional Requirement	Dependency Package
Certification Path Validation – Basic	FDP_CPD.1	none
	FDP_DAU_CPV_INI.1	
	FDP_DAU_CPV_CER.1	
	FDP_DAU_CPV_CER.2	
	FDP_DAU_CPV_OUT.1	
Certification Path Validation – Basic Policy	FDP_DAU_CPV_INI.2	Certification Path Validation – Basic
	FDP_DAU_CPV_OUT.2	
Certification Path Validation – Policy Mapping	FDP_DAU_CPV_INI.3	Certification Path Validation – Basic, Certification Path Validation – Basic Policy
	FDP_DAU_CPV_CER.3	
	FDP_DAU_CPV_OUT.3	
Certification Path Validation – Name Constraints	FDP_DAU_CPV_INI.4	Certification Path Validation – Basic

Package Name	Functional Requirement	Dependency Package
Constraints	FDP_DAU_CPV_CER.4	Validation – Basic
	FDP_DAU_CPV_CER.5	
PKI Signature Generation	FDP_ETC_SIG.1	none
PKI Signature Verification	FDP_ITC_SIG.1	Certification Path Validation – Basic
	FDP_DAU_SIG.1	
PKI Encryption using Key Transfer Algorithms	FDP_ETC_ENC.1	Certification Path Validation – Basic
	FDP_DAU_ENC.1	
PKI Encryption using Key Agreement Algorithms	FDP_ETC_ENC.2	Certification Path Validation – Basic
	FDP_DAU_ENC.2	
PKI Decryption using Key Transfer Algorithms	FDP_ITC_ENC.1	None
PKI Decryption using Key Agreement Algorithms	FDP_ITC_ENC.2	Certification Path Validation – Basic
	FDP_DAU_ENC.3	
PKI Based Entity Authentication	FIA_UAU.1;1	Certification Path Validation – Basic
	FIA_UAU.4	
	FIA_UAU_SIG.1	
	FIA_UID.1;1	
Online Certificate Status Protocol Client	FDP_DAU_OCS.1	None
Certificate Revocation List Validation	FDP_DAU_CRL.1	None
Audit Management	FAU_GEN.1	None
	FAU_GEN.2	
	FAU_SAR.1	
	FAU_SEL.1	
	FAU_STG.1	
Continuous Authentication	FIA_UAU.6:1	PKI Based Entity Authentication, Certification Path Validation – Basic
	FIA_UAU.6:2	

In addition to the above dependencies, the following conditional dependencies may be invoked depending on the selections by the ST author:

- CPV – Basic package may depend on OCSP Client Package

- CPV – Basic package may depend on Certificate Revocation List (CRL) Validation Package
- OCSP Client Package may depend on CPV – Basic package
- Certificate Revocation List (CRL) Validation Package may depend on CPV – Basic package

5.3.1 Certification Path Validation – Basic Package

The functions in this package address the validation of the certification path. Certification path development is also a part of this package. It is realized that the most likely implementations consist of developing a path (using a variety of techniques) and then validating the certification path. It is further recognized that certification path validation generally consists of validating certificates starting with the one certified by a trust anchor and ending with the one issued to the subscriber of interest. However, in order to be implementation neutral, this package does not mandate any ordering of certification path development and certification validation processes. A compliant implementation will only need to meet the security requirements specified in this package.

All processing defined is X.509 and PKIX compliant. The certification path validation in these standards is procedural, but in keeping with the spirit of functional specification, certification path validation requirements are specified using non-procedural techniques.

There are three types of public key certificates:

- Trust anchors: These are self-signed certificates that do not require any validation. The trust anchor (self-signed certificate) is generally in the form of a certificate. The primary purpose of the trust anchor is to obtain the Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable). This package permits validation of trust anchor, including validating signature and verifying that the trust anchor 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 version of this PKE PP family assumes that the path validation is being done as of current time (as opposed to, e.g., verification of old signature in case of dispute). Future versions may include the capability to validate path as of a user-defined time.

If revocation checking is selected, this package may depend on one or both of OCSP Client and CRL validation packages

5.3.1.1 Class FDP – User Data Protection

FDP_CPD.1 Certification path development

Hierarchical to: No other components.

FDP_CPD.1.1 The TSF shall develop a certification path from a trust anchor provided by [selection of one or more by the ST author: *user; administrator,* [assignment by the ST author*: other role defined*]] to the subscriber using matching rules for the following subscriber certificate fields or extensions: [selection of one or more by the ST author: *distinguished name, subject alternative names, subject key identifier, subject public key algorithm, certificate policies,* [assignment by the ST author*: other certificate fields or extensions*]].

FDP_CPD.1.2 The TSF shall develop the certification path using the following additional matching rule: [selection of one by the ST author:

a) none,

b) keyUsage extension has nonRepudiation bit set,

c) keyUsage extension has digitalSignature bit set,

d) keyUsage extension has keyEncipherment bit set,

e) *key Usage extension has keyAgreement bit set*].

FDP_CPD.1.3 The TSF shall develop the certification path using the following additional matching rule [selection of one by the ST author:

a) none,

b) extendedKeyUsage extension contains EFS or anyExtendedKeyUsage OID,

c) extendedKeyUsage extension contains SCL or anyExtendedKeyUsage OID,

d) extendedKeyUsage extension contains code signing or anyExtendedKeyUsage OID,

e) extendedKeyUsage extension contains OCSP signing or anyExtendedKeyUsage OID,

f) [assignment by the ST author*: other extended key usage OID related matching rules*]].

FDP_CPD.1.4 The TSF shall bypass any matching rules except [selection of one or more by the ST author: *distinguished name, subject alternative names, subject key identifier, subject public key algorithm, certificate policies,* [assignment by the ST author*: other certificate fields or extensions*]] if additional certification paths are required.

Dependencies: None

Application Note: *In FDP_CPD.1.2, the assignment nonRepudiation should be used if the path is being developed for signature verification; the assignment digitalSignature should be used if the path is being developed for entity authentication; the assignment keyEncipherment, should be used if the path is being developed for encryption certificate using a key transfer algorithm (e.g., RSA); the assignment keyAgreement should be used if the path is being developed for encryption certificate using a key calculation algorithm (e.g., DH, ECDH).*

In FDP_CPD.1.3, the selection of the matching rule should be made depending on the PKE application requirement. anyExtendedKeyUsage is a match for any application.

FDP_DAU_CPV_INI.1 Certification path initialisation -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_INI.1.1 The TSF shall use the trust anchor provided by [selection of one or more by the ST author: *user, administrator,* [assignment by the ST author*: other role(s) defined*]].

FDP_DAU_CPV_INI.1.2 The TSF shall obtain the current time called “current-time’ from a reliable source [selection of one by the ST author: *local environment,* [assignment by ST author: *other sources defined by ST author*]].

FDP_DAU_CPV_INI.1.3 The TSF shall perform the following checks on the trust anchor [selection of one or more by the ST author:

a) None;

b) Subject DN and Issuer DN match;

c) Signature verifies using the subject public key and parameter (if applicable) from the trust anchor;

d) notBefore field in the trust anchor <= current-time;

e) *notAfter field in the trust anchor => current-time*]

FDP_DAU_CPV_INI.1.4 The TSF shall derive from the trust anchor [selection of one or more by the ST author: *subject DN, subject public key, subject public key algorithm object identifier, subject public key parameters*]

Dependencies: FCS_COP.1, FPT_STM.1

Application Note: *While the PP requires the environment to provide accurate time to required precision, the ST author can choose other sources of accurate time.*

FDP_DAU_CPV_CER.1 Certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_CER.1.1 The TSF shall accept a certificate only if the following checks succeed:

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 < = current-time

c) notAfter field in the certificate > = current-time

d) issuer field in the certificate = parent-DN

e) TSF is able to process all extensions marked critical

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

FDP_DAU_CPV_CER.1.3 The TSF shall bypass the revocation check if the revocation information is not available and [selection of one or more by the ST author: *user, administrator,* [assignment by the ST author*: other role(s) defined*]] overrides revocation checking.

FDP_DAU_CPV_CER.1.4 The TSF shall accept a certificate if the revocation status using [selection of one or more by the ST author: *CRL, OCSP*] demonstrates that the certificate is not revoked.

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

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

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

c) Set parameters = “null”.

Dependencies: FCS_COP.1, FPT_STM.1

Application Note: *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_CER.2 Intermediate certificate processing -- basic

Hierarchical to: No other components.

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

a) basicConstraints field is present with cA = TRUE

b) pathLenConstraint is not violated

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

Dependencies: FDP_DAU_CPV_CER.1

FDP_DAU_CPV_OUT.1 Certification path output -- basic

Hierarchical to: No other components.

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

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

FDP_DAU_CPV_OUT.1.3 The TSF shall output the following additional variables from the end certificate [selection of one or more by the ST author: *certificate, subject alternative names, extendedKeyUsage,* [assignment by the ST author*: other information*]].

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

Dependencies: None

5.3.2 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.

5.3.2.1 Class FDP – User Data Protection

FDP_DAU_CPV_INI.2 Certification path initialisation – basic policy

Hierarchical to: No other components.

FDP_DAU_CPV_INI.2.1 The TSF shall use the initial-certificate-policies provided by [selection of one or more by the ST author: *user, administrator,* [assignment by the ST author*: other role(s) defined*]].

Dependencies: FDP_DAU_CPV_INI.1

FDP_DAU_CPV_OUT.2 Certification path output – basic policy

Hierarchical to: No other components.

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

Dependencies: FDP_DAU_CPV_OUT.1

5.3.3 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.

5.3.3.1 Class FDP – User Data Protection

FDP_DAU_CPV_INI.3 Certification path initialisation – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPV_INI.3.1 The TSF shall use the explicit-policy-indicator, policy-mapping-inhibit-indicator, inhibit-any-policy-indicator provided by [selection of one or more by the ST author: *user, administrator,* [assignment by the ST author*: other role defined*]].

Dependencies: FDP_DAU_CPV_INI.2

FDP_DAU_CPV_CER.3 Intermediate certificate processing – policy mapping

Hierarchical to: No other components.

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

a) explicit-policy-indicator

b) policy-mapping-inhibit-indicator

c) inhibit-any-policy-indicator

Dependencies: FDP_DAU_CPV_CER.2

FDP_DAU_CPV_OUT.3 Certification path output – policy mapping

Hierarchical to: No other components.

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

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

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

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

FDP_DAU_CPV_OUT.3.5 The TSF shall output policy mapping history.

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

Dependencies: FDP_DAU_CPV_OUT.2

5.3.4 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.

5.3.4.1 Class FDP – User Data Protection

FDP_DAU_CPV_INI.4 Certification path initialisation – names

Hierarchical to: No other components.

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

Dependencies: FDP_DAU_CPV_INI.1

FDP_DAU_CPV_CER.4 Certificate processing – name constraints

Hierarchical to: No other components.

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

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

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

c) each hierarchical name form of type [selection of one or more by the ST author: *DN, RFC-822, URL,* [assignment by the ST author*: other hierarchical name forms*]] in the subjectAlternateName field is in at least one of the permitted-subtrees for that name form

d) each hierarchical name form of type [selection of one or more by the ST author: *DN, RFC-822, URL,* [assignment by the ST author*: other hierarchical name forms*]] in the subjectAlternateName field is in none of the excluded-subtrees for that name form

Dependencies: FDP_DAU_CPV_CER.1

FDP_DAU_CPV_CER.5 Intermediate Certificate processing – name constraints

Hierarchical to: No other components.

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

a) permitted-subtrees

b) excluded-subtrees

Dependencies: FDP_DAU_CPV_CER.2

5.3.5 PKI Signature Generation Package

The PKI Signature Generation package uses the private key for signature generation, and provides the ability to generate the signature information.

5.3.5.1 Class FDP – User Data Protection

FDP_ETC_SIG.1 Export of PKI Signature

Hierarchical to: No other component

FDP_ETC_SIG.1.1	The TSF shall use the private to key perform digital signature.
FDP_ETC_SIG.1.2	The TSF shall include the following information with the digital signature [selection of one or more by the ST author: *hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier,* [assignment by the ST author*: other information*]].
Dependencies:	FCS_COP.1

5.3.6 PKI Signature Verification Package

The PKI Signature Verification package processes the signature information, e.g., the PKCS 7 blob, and use the public key to verify a signature. This package is dependent upon the Certification Path Validation – Basic package. The signature verification package uses the Certification Path Validation package data as input.

5.3.6.1 Class FDP – User Data Protection

FDP_ITC_SIG.1 Import of PKI Signature

Hierarchical to no other component

FDP_ITC_SIG.1.1	The TSF shall use the following information from the signed data [selection of one or more by the ST author: *hashing algorithm, signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier,* [assignment by the ST author*: other information*]] during signature verification.
Dependencies:	None

FDP_DAU_SIG.1 Signature Blob Verification

Hierarchical to: No other components.

FDP_DAU_SIG.1.1	The TSF shall use the following information from Certification Path Validation to verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters.
FDP_DAU_SIG.1.2	The TSF shall verify that the keyUsage extension output from the Certification Path Validation has the nonRepudiation bit set.
FDP_DAU_SIG.1.3	The TSF shall apply the following additional checks [selection of one or more by the ST author:

a) *Match the subject DN from the Certification Path Validation with that in the signed data.*

b) *Match the subject alternative name from the Certification Path Validation with that in the signed data.*

c) *Verify that the extendedKeyUsage from Certification Path Validation contains an OID for the PKE application or anyExtendedKeyUsage OID*.

d) [assignment by the ST author: *other checks defined*]].

Dependencies: FCS_COP.1, FDP_DAU_CPV_OUT.1

5.3.7 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.

5.3.7.1 Class FDP – User Data Protection

FDP_ETC_ENC.1 Export of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other component

FDP_ETC_ENC.1.1 The TSF shall include the following information with the encrypted data [selection of one or more by the ST author: *key encryption algorithm, data encryption algorithm, decryptor key identifier,* [assignment by the ST author*: other information*]].

FDP_ETC_ENC.1.2 The TSF shall use the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

Dependencies: FCS_COP.1, FDP_DAU_CPV_OUT.1

FDP_DAU_ENC.1 PKI Encryption Verification – Key Transfer

Hierarchical to: No other components.

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

FDP_DAU_ENC.1.2 The TSF shall apply the following additional checks [selection of one or more by the ST author:

a) *Match the subject DN from the Certification Path Validation with that of the subject of interest.*

b) *Match the subject alternative name from the Certification Path Validation with that of the subject of interest.*

c) *Verify that the extendedKeyUsage from Certification Path Validation contains an OID for the PKE application or anyExtendedKeyUsage OID*.

d) [assignment by the ST author: *other checks defined*]].

Dependencies: FDP_DAU_CPV_OUT.1

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

5.3.8 PKI Encryption using Key Agreement Algorithms Package

This package provides for the performance of public key encryption using key calculation algorithms such as DH or ECDH. Certification path validation is included to ensure that the correct public key of the decrypting party is used. This package is dependent upon the Certification Path Validation – Basic package.

5.3.8.1 Class FDP – User Data Protection

FDP_ETC_ENC.2 Export of PKI Encryption – Key Agreement Algorithms

Hierarchical to: FDP_ETC_ENC.1

FDP_ETC_ENC.2.1 The TSF shall include the following information with the encrypted data [selection of one or more by the ST author: *key encryption algorithm, data encryption algorithm, decryptor key identifier,* [assignment by the ST author*: other information*]].

FDP_ETC_ENC.2.2 The TSF shall use the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

FDP_ETC_ENC.2.3 The TSF shall include the following additional information with the encrypted data [selection of one or more by the ST author: *encryptor public key certificate, encryptor DN, encryptor subject alternative name, encryptor subject key identifier,* [assignment by the ST author*: other information*]].

Dependencies: FCS_COP.1, FDP_DAU_CPV_OUT.1

FDP_DAU_ENC.2 PKI Encryption Verification – Key Agreement, Subject, Decryptor

Hierarchical to: No other components.

FDP_DAU_ENC.2.1 The TSF shall verify that the keyUsage output from Certification Path Validation contains keyAgreement bit set.

FDP_DAU_ENC.2.2 The TSF shall apply the following additional checks [selection of one or more by the ST author:

a) Match the subject DN from the Certification Path Validation with that of the decryptor.

b) *Match the subject alternative name from the Certification Path Validation with that of the decryptor.*

c) *Verify that the extendedKeyUsage from Certification Path Validation is contains the OID for the PKE application or anyExtendedKeyUsage OID*.

d) [assignment by ST author: *other checks defined*]].

Dependencies: FDP_DAU_CPV_OUT.1

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

5.3.9 PKI Decryption using Key Transfer Algorithms Package

This package provides for 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.

5.3.9.1 Class FDP – User Data Protection

FDP_ITC_ENC.1 Import of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other components

FDP_ITC_ENC.1.1 The TSF shall use the following information from the encrypted data [selection of one or more by the ST author: *key encryption algorithm, data encryption algorithm, decryptor key identifier,* [assignment by the ST author*: other information*]] during decryption.

FDP_ITC_ENC.1.2 The TSF shall perform the decryption

Dependencies: FCS_COP.1

5.3.10 PKI Decryption using Key Agreement Algorithms Package

This package provides for the performance of public key decryption using key calculation algorithms such as DH or ECDH. This package is dependent upon the Certification Path Validation – Basic package.

5.3.10.1 Class FDP – User Data Protection

FDP_ITC_ENC.2 Import of PKI Encryption – Key Agreement Algorithms

Hierarchical to: FDP_ITC_ENC.1

FDP_ITC_ENC.2.1 The TSF shall use the following information from the encrypted data [selection of one or more by the ST author: *key encryption algorithm, data encryption algorithm, decryptor key identifier,* [assignment by the ST author*: other information*]] during decryption.

FDP_ITC_ENC.2.2 The TSF shall perform the decryption

FDP_ITC_ENC.2.3 The TSF shall use the following information from Certification Path Validation during decryption: subject public key algorithm, subject public key, subject public key parameters.

FDP_ITC_ENC.2.4 The TSF shall use the following additional information from the encrypted data [selection of one or more by the ST author: *encryptor public key certificate, encryptor DN, encryptor subject alternative name, encryptor subject key identifier,* [assignment by the ST author*: other information*]] during decryption.

Dependencies: FCS_COP.1, FDP_DAU_CPV_OUT.1

FDP_DAU_ENC.3 PKI Encryption Verification – Key Agreement, Subject, Encryptor

Hierarchical to: No other components.

FDP_DAU_ENC.3.1 The TSF shall verify that the keyUsage output from Certification Path Validation contains keyAgreement bit set.

FDP_DAU_ENC.3.2 The TSF shall apply the following additional checks [selection of one or more by the ST author:

a) *Match the subject DN from the Certification Path Validation with that of the encryptor.*

b) *Match the subject alternative name from the Certification Path Validation with that of the encryptor.*

c) *Verify that the extendedKeyUsage from Certification Path Validation contains the OID for the PKE application or anyExtendedKeyUsage OID*.

d) [assignment by the ST author: *other checks defined*]].

Dependencies: FDP_DAU_CPV_OUT.1

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

5.3.11 PKI Based Entity Authentication Package

This package provides for the use of PKI as an entity authentication service. The identification and authentication (I&A) requirements in this package have a different purpose than I&A requirements in base requirements in Section 5.1. The base requirements in Section 5.1 are always required and are used to manage and use the cryptographic keys, whereas this PKI Based Entity Authentication package is used when the PKE application (TOE) performs entity authentication (e.g., Secure Socket Layer (SSL), Transport Layer Security (TLS), etc.). To differentiate between the base I&A requirements and those included in this package, the requirements in this package that are the same as those in the base I&A requirements have been iterated. The following characteristics are valid for either an EAL 3 augmented or EAL 4 augmented PP or ST:

- The ST author should note that this package requires the iteration of certain requirements, including FIA_UAU.1and FIA_UID.1, in order to differentiate

between TOE users and users who are remote entities. The latter users, those who are remote entities, are the specific ones addressed in the requirements below. Note that when this package is selected, one of the roles selected in FMT_SMR.2 in the base requirements will be “remote entity.”

- This package is used to permit the use of a PKI based entity authentication standard for identification and authentication of a remote entity. The standard may or may not determine the authentication failure, selection of secrets, and authentication feedback requirements. Thus, FIA_AFL and FIA_SOS families, and FIA_UAU.7 components were not selected to inclusion in this package.
- This package shall be used for initial authentication of the entity. A dependent package (Continuous Authentication) shall be used for continuous authentication of the protocol, command, packets etc.
- This package only requires a remote entity to authenticate to the TOE. For two-way authentication (e.g., client and server) when each TOE includes the package for authentication of the other, two-way authentication is achieved. In addition, the specification of the standard (e.g., SSL v3) may imply two-way authentication.

This package is dependent upon the Certification Path Validation – Basic package.

5.3.11.1 Class FIA – Identification and Authentication

FIA_UAU.1;1 Timing of authentication – Remote Entity

Hierarchical to: No other components

FIA_UAU.1.1;1	The TSF shall allow [assignment by the ST author*: list of TSF mediated actions*] on behalf of the ***remote entity*** to be performed before the ***remote entity*** is authenticated.
FIA_UAU.1.2;1	The TSF shall require each ***remote entity*** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that ***remote entity***.
Dependencies:	FIA_UID.1 Timing of identification

FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components

FIA_UAU.4.1	The TSF shall prevent reuse of authentication data related to **[selection of one or more by the ST author: *FIPS 196, SSL v2, SSL v3, TLS,* [assignment by the ST author: *other PKI based authentication mechanism(s)*]].**
Dependencies:	None.

FIA_UAU_SIG.1 Entity Authentication

Hierarchical to: No other components.

FIA_UAU_SIG.1.1	The TSF shall use the following information from Certification Path Validation to verify signature on response from the remote entity to the challenge from the TSF: subject public key algorithm, subject public key, subject public key parameters.
FIA_UAU_SIG.1.2	The TSF shall verify that the keyUsage output from Certification Path Validation contains digitalSignature bit set.
FIA_UAU_SIG.1.3	The TSF shall apply the following additional checks [selection of one or more by the ST author: *a) Match the subject DN from the Certification Path Validation with the entity being authenticated.* *b) Match the subject alternative name from the Certification Path Validation with the entity being authenticated.* *c)* [assignment by the ST author: *other checks defined*]].
Dependencies:	FCS_COP.1, FDP_DAU_CPV_OUT.1

FIA_UID.1:1 Timing of identification – Remote Entity

Hierarchical to: No other components

FIA_UID.1.1;1	The TSF shall allow [assignment by the ST author*: list of TSF-mediated actions*] on behalf of the ***remote entity*** to be performed before the ***remote entity*** is identified.
FIA_UID.1.2;1	The TSF shall require each ***remote entity*** to be successfully identified before allowing any other TSF-mediated actions on behalf of that ***remote entity***.
Dependencies:	None.

5.3.12 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. The ST author can assign additional rules to process OCSP extensions. If the OCSP implementation establishes trust in the OCSP responder by performing Certificate Path Validation, then CPV – Basic and other CPV packages may also be applicable, depending upon the implementation.

5.3.12.1 Class FDP – User Data Protection

FDP_DAU_OCS.1 Basic OCSP Client

Hierarchical to: No other component

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

FDP_DAU_OCS.1.2	The OCSP request shall contain the following extensions: [selection of one or more by the ST author: *none, nonce,* [assignment by the ST author: *other extensions*]].
FDP_DAU_OCS.1.3	The TSF shall obtain the public key, algorithm, and public key parameters of the OCSP Responder from [selection of one by the ST author: *trust anchor, certificate signing CA, OCSP responder certificate,* [assignment by ST author: *other sources*]].
FDP_DAU_OCS.1.4	The TSF shall perform the following additional function [selection of one by the ST author: *a) none; or* b) *establish trust in OCSP responder certificate using* [selection of one or more by the ST author: *certification path validation – basic, certification path validation – basic policy, certification path validation –policy mapping, certification path validation – name constraint*]].
FDP_DAU_OCS.1.5	The TSF shall verify signature on the OCSP response using trusted public key, algorithm, and public key parameters of the OCSP responder.
FDP_DAU_OCS.1.6	The TSF shall verify that if the OCSP responder certificate contains extendedKeyUsage extension, the extension contains the PKIX OID for ocsp-signing or the anyExtendedKeyUsage OID.
FDP_DAU_OCS.1.7	The TSF shall match the responderID in the OCSP response with the corresponding information in the responder certificate
FDP_DAU_OCS.1.8	The TSF shall match the certID in a request with certID in singleResponse.
FDP_DAU_OCS.1.9	The TSF shall accept the OCSP response for all entries as current if the following policy is met: [selection of one by the ST author: *always, current-time <= producedAt + x where x is provided by* [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]]].
FDP_DAU_OCS.1.10	The TSF shall accept the OCSP response for an entry as current if the following policy is met: [selection of one by the ST author: *always, current-time <= thisUpdate for entry + x where x is provided by* [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]]].
FDP_DAU_OCS.1.11	The TSF shall accept the OCSP response for an entry as current if the following policy is met: [selection of one by the ST author: *always, current-time <= nextUpdate for entry + x where x is provided by* [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]]].
FDP_DAU_OCS.1.12	The TSF shall accept OCSP response as current if [selection of one or more by the ST author: *user, administrator,* [assignment by

the ST author: *other role(s) defined*]] overrides freshness checking.

FDP_DAU_OCS.1.13 The TSF shall reject OCSP response if the response contains "critical" extension(s) that TSF does not process.

FDP_DAU_OCS.1.14 The TSF shall perform the following additional checks [selection of one or more by the ST author:

a) none,

b) request nonce = response nonce,

c) [assignment by ST author: *other rule(s)*]].

Dependencies: FCS_COP.1, FPT_STM.1

5.3.13 Certificate Revocation List (CRL) Validation Package

This package is used for validating a CRL. This version of the document does not require processing of CRL issuing distribution point (IDP) CRL or delta CRL. Future versions may include that capability by codifying Annex B of X.509 standard.

It should be noted that this package may be used to process a CRL that is pointed to by a CRL Distribution Point (CRLDP) extension in a certificate as long as the CRL is a full CRL, indicated by the absence of IDP and deltaCRLIndicator extensions.

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. If the compliant implementation develops a certification path, then a certification path validation package may also be applicable.

The ST author can assign additional rules to process Issuing Distribution Point CRL and Delta CRL.

5.3.13.1 Class FDP – User Data Protection

FDP_DAU_CRL.1 Basic CRL Checking

Hierarchical to no other component

FDP_DAU_CRL.1.1 The TSF shall obtain the CRL from [selection of one or more by the ST author: *local cache, repository, location point to by the CRL DP in public key certificate of interest, user*, [assignment: *other locations defined by the ST author*]].

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

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

FDP_DAU_CRL.1.4	The TSF shall verify that if a critical keyUsage extension is present in CRL issuer certificate, cRLSign bit in the extension is set in the certificate.
FDP_DAU_CRL.1.5	The TSF shall match the issuer field in the CRL with what it assumes to be the CRL issuer.
FDP_DAU_CRL.1.6	The TSF shall accept the CRL as current if the following policy is met: [selection of one by the ST author: *always, current-time <= thisUpdate + x where x is provided by* [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]]].
FDP_DAU_CRL.1.7	The TSF shall accept the CRL as current if the following policy is met: [selection of one by the ST author: *always, current-time <= nextUpdate + x where x is provided by* [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]]].
FDP_DAU_CRL.1.8	The TSF shall accept CRL as current if [selection by the ST author: *user, administrator,* [assignment by the ST author: *other role(s) defined*]] overrides freshness checking.
FDP_DAU_CRL.1.9	The TSF shall reject CRL if the CRL contains “critical” extension(s) that TSF does not process.
FDP_DAU_CRL.1.10	The TSF shall perform the following additional checks [selection of one or more by the ST author: *a) none,* b) [assignment by ST author: *other rule(s)*]].
Dependencies:	FCS_COP.1, FPT_STM.1
Application Note:	*The trusted public key, algorithm, and public key parameters of the CRL issuer should normally be the same as those used for verifying signature on the certificate being checked for revocation. If not, at least certificate path development – basic can be used to obtain the public key.*

5.3.14 Audit Management Package

This package is used in order to generate and protect audit events relevant to the PKE applications (TOEs). Examples of PKE application audit events are:

- Management of trust anchors (addition, deletion)
- Identification and Authentication
- Signature verification success, date and time, and policies under which signatures were valid
- Signature verification failure, date and time, cause of failure (signature on the object failed, certification path failure, policy failure, etc.)

- User override events (current CRL availability, accept policy failure, accept null policy, accept other policy, etc.)

The security functional requirements below provide an accurate and complete list of auditable events.

Some or all of the requirements for the Audit Management package can be met by environment such as trusted operating system. For example, the protection of the audit data may be satisfied by the host operating system. Alternatively, the audit data may be protected by the access control policy components or TSF data protection related components in the base requirements.

Also, many of the dependencies for this package are satisfied by the base TOE security functional requirements or environmental requirements. Examples of these dependencies include:

- Reliable time stamp – Provided as a part of security functional requirements for the IT Environment
- User identification – Provided as a part of base TOE security functional requirements

The Rationale section of this PP family provides accurate and complete dependency analysis.

5.3.14.1 Class FAU – Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other component

FAU_GEN.1.1 The TSF 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 for the [selection **of one** by the ST author: *minimum, basic, detailed, not specified*] level of audit; and

c) **[selection of one or more by the ST author:**

- ***a) Use of identification & authentication (success and failure)***
- ***b) Use of private key***
- ***c) Attempt to bypass access control policy***
- ***d) Path development failure***
- ***e) Path validation failure***
- ***f) Trust anchor check failure***
- ***g) Certificate check failure***
- ***h) Signature verification failure***

*i) **Signature verification success***

*j) **Modification, deletion and other changes to trust anchors***

*k) **Changes to security attributes***

*l) **Changes to default values of security attributes***

*m) **CRL processing failure***

*n) **OCSP response processing failure***

*o) **User override of any failure such as CRL, OCSP, certificate, certification path, etc.***

p) **[assignment by the ST author: *other specifically defined auditable events*]]**.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, 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, **[selection of one or more by the ST author: *reason for failure, valid policies, old security attribute value, new security attribute value, hash of trust anchor, key identifier, identity of data signer,* [assignment by the ST author: *other audit relevant information*]]**.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1 The TSF 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 TSF shall provide [assignment by the ST author: *authorised users*] with the capability to read **[selection of one or more by the ST author: *all audit information,* [assignment by the ST author: *list of audit information*]]** from the audit records.

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

Dependencies: FAU_GEN.1 Audit data generation

FAU_SEL.1	**Selective audit**
Hierarchical to:	No other components.
FAU_SEL.1.1	The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: a) [selection of **one or more by the ST author**: *object identity, user identity, subject identity, host identity, event type*] b) **[selection of one or more by the ST author: *signer identity, policy identifier, key identifier, trust anchor,* [assignment by the ST author: *list of additional attributes that audit selectivity is based upon*]].**
Dependencies:	FAU_GEN.1 Audit data generation FMT_MTD.1 Management of TSF data
FAU_STG.1	**Protected audit trail storage**
Hierarchical to:	No other components.
FAU_STG.1.1	The TSF shall protect the stored audit records from unauthorised deletion.
FAU_STG.1.2	The TSF shall be able to [selection by the ST author: *prevent, detect*] modifications to the audit records.
Dependencies:	FAU_GEN.1 Audit data generation

5.3.15 Continuous Authentication Package

This package provides for the use of the continuous authentication service of remote entity. This package is dependent on the PKI Based Entity Authentication Package and the CPV – Basic package. This package is used for continuous authentication of remote entity. The following characteristics are valid for either an EAL 3 augmented or EAL 4 augmented PP or ST:

- The ST author should note that this package requires the iteration of certain FIA_UAU.6 in order to differentiate between TOE users and users who are remote entities. The latter users, those who are remote entities, are the specific ones addressed in the requirements below.
- This package only requires a remote entity to authenticate to the TOE. For two-way authentication (e.g., client and server) when each TOE includes the package for authentication of the other, two-way authentication is achieved. In addition, the specification of the standard (e.g., SSL v3) may imply two-way authentication.

5.3.15.1 Class FIA – Identification and Authentication

FIA_UAU.6:1 Re-authenticating remote entity

Hierarchical to: No other components

FIA_UAU.6.1;1	The TSF shall re-authenticate the ***remote entity*** under the conditions **[selection of one or more by the ST author: *each packet, each command, each transaction,* [assignment by ST author: *list of conditions under which re-authentication is required*]]**.
Dependencies:	None.
Application Note:	*It is acceptable to use the symmetric session cryptographic key established during the initial authentication in conjunction with integrity and authentication functions such as HMAC for re-authentication of commands, packets, transactions, etc.*

FIA_UAU.6:2 Re-authenticating user

Hierarchical to: No other components

FIA_UAU.6.1;2	The TSF shall re-authenticate the user under the conditions **[selection of one or more by the ST author: *none, each packet, each command, each transaction,* [assignment by ST author: *list of conditions under which re-authentication is required*]]**.
Dependencies:	None.
Application Note:	*Selection of none means that this component is used for only initial authentication and not for continuous authentication.* *It is acceptable to use the symmetric session cryptographic key established during the initial authentication in conjunction with integrity and authentication functions such as HMAC for re-authentication of commands, packets, transactions, etc.*

5.4 PPs With EAL 3 With Augmentation

The PP/ST author may select assurance components of Evaluation Assurance Level 3 (EAL3) augmented by ALC_FLR.1. All requirements are drawn from Part 3 of the Common Criteria. The assurance components are listed in Table 5.4. EAL 3 with augmentation will be selected when the TOE requires a moderate level of independently assured security and requires a thorough investigation of the TOE and its development without substantial re-engineering.

Table 5.4 – EAL3 with Augmentation Assurance Requirements

Assurance Component Identifier	**Assurance Component Title**
ACM_CAP.3	Authorisation controls
ACM_SCP.1	TOE CM coverage
ADO_DEL.1	Delivery procedures
ADO_IGS.1	Installation, generation, and start-up procedures
ADV_FSP.1	Informal functional specification
ADV_HLD.2	Security enforcing high-level design
ADV_RCR.1	Informal correspondence demonstration
AGD_ADM.1	Administrator guidance
AGD_USR.1	User guidance
ALC_DVS.1	Identification of security measures
ALC_FLR.1	Basic flaw remediation
ATE_COV.2	Analysis of coverage
ATE_DPT.1	Testing: high-level design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing - sample
AVA_MSU.1	Examination of guidance
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.1	Developer vulnerability analysis

5.4.1 Class ACM: Configuration Management

ACM_CAP.3 Authorisation controls

Dependencies: ALC_DVS.1 Identification of security measures

Developer action elements:

ACM_CAP.3.1D	The developer shall provide a reference for the TOE.
ACM_CAP.3.2D	The developer shall use a CM system.
ACM_CAP.3.3D	The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.3.1C	The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.3.2C	The TOE shall be labeled with its reference.
ACM_CAP.3.3C	The CM documentation shall include a configuration list and a CM plan.
ACM_CAP.3.NEWC	The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.3.4C	The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.3.5C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.3.6C	The CM system shall uniquely identify all configuration items.
ACM_CAP.3.7C	The CM plan shall describe how the CM system is used.
ACM_CAP.3.8C	The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.3.9C	The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.3.10C	The CM system shall provide measures such that only authorised changes are made to the configuration items.

Evaluator action elements:

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

ACM_SCP.1 TOE CM Coverage

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ACM_SCP.1.1D	The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements:

ACM_SCP.1.1C	The list of configuration items shall include the following: implementation representation and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements:

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

5.4.2 Class ADO: Delivery and Operation

ADO_DEL.1 Delivery Procedures

Dependencies: No dependencies.

Developer action elements:

ADO_DEL.1.1D	The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2D	The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

Evaluator action elements:

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

ADO_IGS.1 Installation, Generation, and Start-up Procedures

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

ADO_IGS.1.1D	The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C	The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E	The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.4.3 Class ADV: Development

ADV_FSP.1 Informal functional specification

Dependencies: ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_FSP.1.1D	The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.1.1C	The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2C	The functional specification shall be internally consistent.
ADV_FSP.1.3C	The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4C	The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV_FSP.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 Security enforcing high-level design

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.2.1D	The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.2.1C	The presentation of the high-level design shall be informal.
ADV_HLD.2.2C	The high-level design shall be internally consistent.
ADV_HLD.2.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5C	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.2.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7C	The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8C	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.2.9C	The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV_HLD.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Dependencies: No dependencies.

Developer action elements:

ADV_RCR.1.1D	The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C	For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

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

5.4.4 Class AGD: Guidance Documents

AGD_ADM.1Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_ADM.1.1D	The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C	The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C	The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C	The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5C	The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6C	The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C	The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C	The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D	The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C	The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C	The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C	The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C	The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5C	The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6C	The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

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

5.4.5 Class ALC: Life Cycle Support

ALC_DVS.1 Identification of security measures

Dependencies: No dependencies.

Developer action elements:

ALC_DVS.1.1D	The developer shall produce development security documentation.

Content and presentation of evidence elements:

ALC_DVS.1.1C	The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C	The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

ALC_DVS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2E	The evaluator shall confirm that the security measures are being applied.

ALC_FLR.1 Basic flaw remediation

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.1.1D	The developer shall document the flaw remediation procedures.

Content and presentation of evidence elements:

ALC_FLR.1.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

Evaluator action elements:

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

5.4.6 Class ATE: Tests

ATE_COV.2 Analysis of coverage

Dependencies:	ADV_FSP.1 Informal functional specification, ATE_FUN.1 Functional testing
Developer action elements:	
ATE_COV.2.1D	The developer shall provide an analysis of the test coverage.
Content and presentation of evidence elements:	
ATE_COV.2.1C	The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.2.2C	The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
Evaluator action elements:	
ATE_COV.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.1 Testing: high-level design

Dependencies:	ADV_HLD.1 Descriptive high-level design, ATE_FUN.1 Functional testing
Developer action elements:	
ATE_DPT.1.1D	The developer shall provide the analysis of the depth of testing.
Content and presentation of evidence elements:	
ATE_DPT.1.1C	The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
Evaluator action elements:	
ATE_DPT.1.2E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Dependencies:	No dependencies.
Developer action elements:	
ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C	The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3C	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.1.4C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C	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.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent Testing - Sample

Dependencies:	ADV_FSP.1 Informal functional specification, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance, ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D	The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer’s functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.4.7 Class AVA: Vulnerability Assessment

AVA_MSU.1 Examination of guidance

Dependencies:	ADO_IGS.1 Installation, generation, and start-up procedures, ADV_FSP.1 Informal functional specification, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.1.1D	The developer shall provide guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.1.1C	The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.1.2C	The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.1.3C	The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.1.4C	The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements:

AVA_MSU.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.1.2E	The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.1.3E	The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_HLD.1 Descriptive high-level design

Developer action elements:

AVA_SOF.1.1D	The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C	For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C	For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E	The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1 Developer vulnerability analysis

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_HLD.1 Descriptive high-level design, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance

Developer action elements:

AVA_VLA.1.1D	The developer shall perform a vulnerability analysis.
AVA_VLA.1.2D	The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements:

AVA_VLA.1.1C	The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.
AVA_VLA.1.2C	The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.
AVA_VLA.1.3C	The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements:

AVA_VLA.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.1.2E	The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

5.5 PPs With EAL 4 With Augmentation

The PP/ST author may select the assurance components of Evaluation Assurance Level 4 (EAL4) augmented by ALC_FLR.1. All requirements are drawn from Part 3 of the Common Criteria. The assurance components are listed in Table 5.5. EAL 4 with augmentation will be selected in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs.

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.1 is added to provide basic flaw remediation.

Table 5.5 – EAL4 with Augmentation Assurance Requirements

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

Assurance Component Identifier	Assurance Component Title
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing – sample
AVA_MSU.2	Validation of analysis
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.2	Independent vulnerability analysis

5.5.1 Class ACM: Configuration management

ACM_AUT.1 Partial CM automation

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

Content and presentation of evidence elements:

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system**.**

Evaluator action elements:

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

ACM_CAP.4 Generation support and acceptance procedures

Dependencies: ALC_DVS.1 Identification of security measures

Developer action elements:

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.4.1C	The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.4.2C	The TOE shall be labelled with its reference.
ACM_CAP.4.3C	The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
ACM_CAP.4.NEWC	The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.4C	The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.5C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.4.6C	The CM system shall uniquely identify all configuration items.
ACM_CAP.4.7C	The CM plan shall describe how the CM system is used.
ACM_CAP.4.8C	The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.4.9C	The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.4.10C	The CM system shall provide measures such that only authorised changes are made to the configuration items.
ACM_CAP.4.11C	The CM system shall support the generation of the TOE.
ACM_CAP.4.12C	The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

ACM_CAP.4.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.2 Problem tracking CM coverage

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ACM_SCP.2.1D	The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements:

ACM_SCP.2.1C	The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements:

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

5.5.2 Class ADO: Delivery and operation

ADO_DEL.2 Detection of modification

Dependencies: **ACM_CAP.3 Authorisation controls**

Developer action elements:

ADO_DEL.2.1D	The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.2.2D	The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.2.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.2.2C	The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
ADO_DEL.2.3C	The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Evaluator action elements:

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

ADO_IGS.1 Installation, generation, and start-up procedures

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

ADO_IGS.1.1D	The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C	The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E	The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.5.3 Class ADV: Development

ADV_FSP.2 Fully defined external interfaces

Dependencies:	ADV_RCR.1 Informal correspondence demonstration
Developer action elements:	
ADV_FSP.2.1D	The developer shall provide a functional specification.
Content and presentation of evidence elements:	
ADV_FSP.2.1C	The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2C	The functional specification shall be internally consistent.
ADV_FSP.2.3C	The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4C	The functional specification shall completely represent the TSF.
ADV_FSP.2.5C	The functional specification shall include rationale that the TSF is completely represented.
Evaluator action elements:	
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 Security enforcing high-level design

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_RCR.1 Informal correspondence demonstration
Developer action elements:	
ADV_HLD.2.1D	The developer shall provide the high-level design of the TSF.
Content and presentation of evidence elements:	
ADV_HLD.2.1C	The presentation of the high-level design shall be informal.
ADV_HLD.2.2C	The high-level design shall be internally consistent.
ADV_HLD.2.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5C	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.2.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7C	The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8C	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.2.9C	The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
Evaluator action elements:	
ADV_HLD.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.1 Subset of the Implementation of the TSF

Dependencies:	ADV_LLD.1 Descriptive low-level design, ADV_RCR.1 Informal correspondence demonstration, ALC_TAT.1 Well-defined development tools
Developer action elements:	
ADV_IMP.1.1D	The developer shall provide the implementation representation for a selected subset of the TSF.
Content and presentation of evidence elements:	
ADV_IMP.1.1C	The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2C	The implementation representation shall be internally consistent.
Evaluator action elements:	
ADV_IMP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.2.2E	The evaluator shall determine that the least abstract TSF representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV_LLD.1 Descriptive low-level design

Dependencies:	ADV_HLD.2 Security enforcing high-level design, ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_LLD.1.1D	The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

ADV_LLD.1.1C	The presentation of the low-level design shall be informal.
ADV_LLD.1.2C	The low-level design shall be internally consistent.
ADV_LLD.1.3C	The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4C	The low-level design shall describe the purpose of each module.
ADV_LLD.1.5C	The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
ADV_LLD.1.6C	The low-level design shall describe how each TSP-enforcing function is provided.
ADV_LLD.1.7C	The low-level design shall identify all interfaces to the modules of the TSF.
ADV_LLD.1.8C	The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
ADV_LLD.1.9C	The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_LLD.1.10C	The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV_LLD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_LLD.1.2E	The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Dependencies:	No dependencies.

Developer action elements:

ADV_RCR.1.1D	The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C	For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

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

ADV_SPM.1 Informal TOE security policy model

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

ADV_SPM.1.1D	The developer shall provide a TSP model.
ADV_SPM.1.2D	The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C	The TSP model shall be informal.
ADV_SPM.1.2C	The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
ADV_SPM.1.3C	The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
ADV_SPM.1.4C	The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

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

5.5.4 Class AGD: Guidance Documents

AGD_ADM.1 Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_ADM.1.1D	The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C	The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C	The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C	The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5C	The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6C	The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C	The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C	The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D	The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C	The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C	The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C	The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C	The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5C	The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6C	The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

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

5.5.5 Class ALC: Life cycle support

ALC_DVS.1 Identification of security measures

Dependencies: No dependencies.

Developer action elements:

ALC_DVS.1.1D	The developer shall produce development security documentation.

Content and presentation of evidence elements:

ALC_DVS.1.1C	The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C	The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

ALC_DVS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2E	The evaluator shall confirm that the security measures are being applied.

ALC_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC_LCD.1.1D	The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2D	The developer shall provide life-cycle definition documentation.

Content and presentation of evidence elements:

ALC_LCD.1.1C	The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2C	The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

Evaluator action elements:

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

ALC_FLR.1 Basic flaw remediation

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.1.1D	The developer shall document the flaw remediation procedures.

Content and presentation of evidence elements:

ALC_FLR.1.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

Evaluator action elements:

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

ALC_TAT.1 Well-defined development tools

Dependencies: ADV_IMP.1 Subset of the implementation of the TSF

Developer action elements:

ALC_TAT.1.1D	The developer shall identify the development tools being used for the TOE.
ALC_TAT.1.2D	The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C	All development tools used for implementation shall be well-defined.
ALC_TAT.1.2C	The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.
ALC_TAT.1.3C	The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

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

5.5.6 Class ATE: Tests

ATE_COV.2 Analysis of coverage

Dependencies:	ADV_FSP.1 Informal functional specification, ATE_FUN.1 Functional testing
Developer action elements:	
ATE_COV.2.1D	The developer shall provide an analysis of the test coverage.
Content and presentation of evidence elements:	
ATE_COV.2.1C	The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.2.2C	The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
Evaluator action elements:	
ATE_COV.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.1 Testing: high-level design

Dependencies:	ADV_HLD.1 Descriptive high-level design
	ATE_FUN.1 Functional testing
Developer action elements:	
ATE_DPT.1.1D	The developer shall provide the analysis of the depth of testing.
Content and presentation of evidence elements:	
ATE_DPT.1.1C	The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
Evaluator action elements:	
ATE_DPT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Dependencies:	No dependencies.
Developer action elements:	
ATE_FUN.1.1D	The developer shall test the TSF and document the results.

ATE_FUN.1.2D	The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C	The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3C	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.1.4C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C	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.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing – sample

Dependencies:	ADV_FSP.1 Informal functional specification, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance, ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D	The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.5.7 Class AVA: Vulnerability Assessment

AVA_MSU.2 Validation of analysis

Dependencies:	ADO_IGS.1 Installation, generation, and start-up procedures, ADV_FSP.1 Informal functional specification, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.2.1D	The developer shall provide guidance documentation.
AVA_MSU.2.2D	The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C	The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.2.2C	The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.2.3C	The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.2.4C	The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.2.5C	The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

AVA_MSU.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.2.2E	The evaluator shall repeat all configuration and installation procedures**,** and other procedures selectively**,** to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.2.3E	The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.
AVA_MSU.2.4E	The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_HLD.1 Descriptive high-level design

Developer action elements:

AVA_SOF.1.1D	The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C	For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2C	For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E	The evaluator shall confirm that the strength claims are correct.

AVA_VLA.2 Independent vulnerability analysis

Dependencies:	ADV_FSP.1 Informal functional specification, ADV_HLD.2 Security enforcing high-level design, ADV_IMP.1 Subset of the implementation of the TSF, ADV_LLD.1 Descriptive low-level design, AGD_ADM.1 Administrator guidance, AGD_USR.1 User guidance

Developer action elements:

AVA_VLA.2.1D	The developer shall perform a vulnerability analysis.
AVA_VLA.2.2D	The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements:

AVA_VLA.2.1C	The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.
AVA_VLA.2.2C	The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.
AVA_VLA.2.3C	The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.
AVA_VLA.2.4C	The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

Evaluator action elements:

AVA_VLA.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.2.2E	The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.
AVA_VLA.2.3E	The evaluator shall perform an independent vulnerability analysis.
AVA_VLA.2.4E	The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.
AVA_VLA.2.5E	The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

6 Rationale

This section provides further evidence and explanation to support the certification of this family of PPs.

6.1 Security Objectives Rationale

6.1.1 Base and Environmental Security Objectives Rationale

Table 6.1 maps base assumptions and threats to objectives, demonstrating that all assumptions and threats are mapped to at least one objective. Table 6.2 maps base objectives to threats and assumptions, demonstrating that all objectives are mapped to at least one threat or assumption.

Table 6.1 – Mapping the TOE Base Assumptions and Threats to Objectives

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

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

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

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

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

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

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

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

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

AE.PKI_Info states that the certificate and certificate revocation information is available to the TOE. AE.PKI_Info is mapped to:

- **OE.PKI_Info**, which states that the IT environment shall provide the TOE certificate and certificate revocation information.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In Table 6.2, the Base TOE and Environmental Objectives are mapped back to threats and assumptions, thereby demonstrating that every objective is mapped to a threat or assumption. Explanation of the mapping is defined above and is not repeated following Table 6.2. Note, once again, these threats and objectives are included in every PP in this PP family.

Table 6.2 – Mapping the Base TOE and Environmental Objectives to Threats and Assumptions

Objective	Threats
OE.Authorized_Users	AE.Authorized_Users
OE.Configuration	AE.Configuration
OE.Crypto	AE.Crypto_Module
OE.Low	AE.Low
OE.Physical_Security	AE.Physical_Protection
OE.Time	AE.Time
O.DAC	T.Attack, T.Modify, T.Private_key
O.I&A	T.Imperson
O.Init_Secure_Attr	T.Object_Init
O.Invoke	T.Bypass
O.Limit_Actions_Auth	T.Imperson
O.Limit_Tries	T.Tries
O.No_Echo	T.Shoulder_Surf
O.Protect_I&A_Data	T.Modify
O.Secure_Attributes	T.Secure_Attributes
O.Security_Roles	T.Role
O.Self_Protect	T.Modify

Objective	Threats
O.Trust_Anchor	T.Modify
O.TSF_Data	T.Modify

6.1.2 Security Objectives Rationale for Packages

The following subsections provide the mapping and rationale for the security objectives and threats associated with each individual package.

6.1.2.1 CPV – Basic Package Security Objectives Rationale

The following tables demonstrate the mapping of threats to objectives and objectives to threats for the CPV – Basic package. Explanatory text is provided below the tables to support the mapping.

Table 6.3 – Mapping of Threats to Objectives for 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_Time O.Current_Certificate
4	T.Masquarade	O.Trusted_Keys
5	T.No_Crypto	O.Get_KeyInfo
6	T.Path_Not_Found	O.Path_Find
7	T.Revoked_Certificate	O.Valid_Certificate
8	T.User_CA	O.User

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 could be used for signature verification. This threat is mapped to:

- **O.Correct_Time**, 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.

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

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

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

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

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

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

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

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

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

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

Table 6.4 maps objectives for the CPV – Basic Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.4.

Table 6.4 – Mapping of Objectives to Threats for CPV – Basic Package

#	Objective	Threats
1	O.Availability	T.DOS_CPV_Basic
2	O.Correct_Time	T.Expired_Certificate
3	O.Current_Certificate	T.Expired_Certificate
4	O.Get_KeyInfo	T.No_Crypto
5	O.Path_Find	T.Path_Not_Found
6	O.Trusted_Keys	T.Masquarade

#	Objective	Threats
7	O.User	T.User_CA
8	O.Verified_Certificate	T.Certificate_Modi
9	O.Valid_Certificate	T.Revoked_Certificate

6.1.2.2 CPV – Basic Policy Package Security Objectives Rationale

The mapping of threats to objectives for the CPV – Basic Policy package is shown in Table 6.5. Text that further supports for the mapping is provided following Table 6.5.

Table 6.5 – Mapping of Threats to Objectives for CPV – Basic Policy Package

#	Threat	Objectives
1	T.Unknown_Policies	O.Provide_Policy_Info

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.

Table 6.6 maps objectives for the CPV – Basic Policy package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.6.

Table 6.6 – Mapping of Objectives to Threats for CPV – Basic Policy Package

#	Objective	Threats
1	O.Provide_Policy_Info	T.Unknown_Policies

6.1.2.3 CPV –Policy Mapping Package Security Objectives Rationale

The mapping of threats to objectives for the CPV – Policy Mapping package is shown in Table 6.7. Text that further supports for the mapping is provided following Table 6.7.

Table 6.7 – Mapping of Threats to Objectives for CPV – Policy Mapping Package

#	Threat	Objectives
1	T.Mapping	O.Map_Policies
2	T.Wrong_Policy_Dec	O.Policy_Enforce

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.

Table 6.8 maps objectives for the CPV – Policy Mapping package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.8.

Table 6.8 – Mapping of Objectives to Threats for CPV – Policy Mapping Package

#	Objective	Threats
1	O.Map_Policies	T.Mapping
2	O.Policy_Enforce	T.Wrong_Policy_Dec

6.1.2.4 CPV – Name Constraints Package Security Objectives Rationale

The mapping of threats to objectives for the CPV – Name Constraints Package is shown in Table 6.9. Text that further supports for the mapping is provided following Table 6.9.

Table 6.9 – Mapping of Threats to Objectives for CVP – Name Constraints Package

#	Threat	Objectives
1	T.Name_Collision	O.Authorised_Names

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

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

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

Table 6.10 – Mapping of Objectives to Threats for CPV – Name Constraints Package

#	Objective	Threats
1	O.Authorised_Names	T.Name_Collision

6.1.2.5 PKI Signature Generation Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Signature Generation package is shown in Table 6.11. Text that further supports for the mapping is provided following Table 6.11.

Table 6.11 – Mapping of Threats to Objectives for the PKI Signature Generation Package

#	Threat	Objectives
1	T.Clueless_PKI_Sig	O.Give_Sig_Hints

T.Clueless_PKI_Sig states that the user may try only inappropriate certificates for PKI signature in the absence of 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.

Table 6.12 maps objectives for the PKI Signature Generation package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.12.

Table 6.12 – Mapping of Objectives to Threats for the PKI Signature Generation Package

#	Objective	Threats
1	O.Give_Sig_Hints	T.Clueless_PKI_Sig

6.1.2.6 PKI Signature Verification Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Signature Verification package is shown in Table 6.13. Text that further supports for the mapping is provided following Table 6.13.

Table 6.13 – Mapping of Threats to Objectives for the PKI Signature Verification Package

#	Threat	Objectives
1	T.Assumed_Identity_PKI_Ver	O.Linkage_Sig_Ver
2	T.Clueless_PKI_Ver	O.Use_Sig_Hints

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 in the absence of a hint. 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.

Table 6.14 maps objectives The PKI Signature Verification package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.14.

Table 6.14 – Mapping of Objectives to Threats for the PKI Signature Verification Package

#	Objective	Threats
1	O.Use_Sig_Hints	T.Clueless_PKI_Ver
2	O.Linkage_Sig_Ver	T.Assumed_Identity_PKI_Ver

6.1.2.7 PKI Encryption using Key Transfer Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for all of PKI Encryption using Key Transfer Algorithms package is shown in Table 6.15. Text that further supports for the mapping is provided following Table 6.15.

Table 6.15 – Mapping of Threats to Objectives for the PKI Encryption using Key Transfer Algorithms Package

#	Threat	Objectives
1	T.Assumed_Identity_WO_En	O.Linkage_Enc_WO
2	T.Clueless_WO_En	O.Hints_Enc_WO

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.

Table 6.16 maps objectives for the PKI Encryption using Key Transfer Algorithms package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.16.

Table 6.16 – Mapping of Objectives to Threats for the PKI Encryption using Key Transfer Algorithms Package

#	Objective	Threats
1	O.Hints_Enc_WO	T.Clueless_WO_En
2	O.Linkage_Enc_WO	T.Assumed_Identity_WO_En

6.1.2.8 PKI Encryption using Key Agreement Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Encryption using Key Agreement Algorithms package is shown in Table 6.17. Text that further supports for the mapping is provided following Table 6.17.

Table 6.17 – Mapping of Threats to Objectives for PKI Encryption using Key Agreement Algorithms Package

#	Threat	Objectives
1	T.Assumed_Identity_With_En	O.Linkage_Enc_W
2	T.Clueless_With_En	O.Hints_Enc_W

T.Assumed_Identity_With_En states that a user may assume the identity of another user to perform encryption using Key Agreement Algorithms. This threat is addressed by:

- **O.Linkage_Enc_W**, which states that the TSF shall use the correct user public key for key agreement during encryption.

T.Clueless_With_En states that the user may try only inappropriate certificates for PKI Encryption using Key Agreement algorithms in absence of hint. This threat is addressed by:

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

Table 6.18 maps objectives for the PKI Encryption using Key Agreement Algorithms package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.18.

Table 6.18 – Mapping of Objectives to Threats for PKI Encryption using Key Agreement Algorithms Package

#	Objective	Threats
1	O.Hints_Enc_W	T.Clueless_With_En
2	O.Linkage_Enc_W	T.Assumed_Identity_With_En

6.1.2.9 PKI Decryption using Key Transfer Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Decryption using Key Transfer Algorithms package is shown in Table 6.19. Text that further supports for the mapping is provided following Table 6.19.

Table 6.19 – Mapping of Threats to Objectives for the PKI Decryption using Key Transfer Algorithms Package

#	Threat	Objectives
1	T.Garble_WO_De	O.Correct_KT

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.

Table 6.20 maps objectives for the PKI Decryption using Key Transfer Algorithms package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.20.

Table 6.20 – Mapping of Objectives to Threats for the PKI Decryption using Key Transfer Algorithms Package

#	Objective	Threats
1	O.Correct_KT	T.Garble_WO_De

6.1.2.10 PKI Decryption using Key Agreement Algorithms Package Security Objectives Rationale

The mapping of threats to objectives for the PKI Decryption using Key Agreement Algorithms package is shown in Table 6.21. Text that further supports for the mapping is provided following Table 6.21.

Table 6.21 – Mapping of Threats to Objectives for PKI Decryption using Key Agreement Algorithms Package

#	Threat	Objectives
1	T.Assumed_Identity_With_De	O.Linkage_Dec_W
2	T.Clueless_With_De	O.Hints_Dec_W
3	T.Garble_With_De	O.Correct_KA

T.Assumed_Identity_With_De states that a user may assume the identity of another user to perform PKI decryption using Key Agreement algorithms. This threat is addressed by:

- **O.Linkage_Dec_W**, which states that the TSF shall use the correct user public key for key agreement during decryption.

T.Clueless_With_De states that the user may try only inappropriate certificates in absence of hint to perform PKI decryption using Key Agreement algorithms. This threat is addressed by:

- **O.Hints_Dec_W**, which states that the TSF shall provide hints for selecting correct certificates or keys for PKI decryption using Key Agreement algorithms.

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

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

Table 6.22 maps objectives for the PKI Decryption With DH, ECDH package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.22.

Table 6.22 – Mapping of Objectives to Threats for PKI Decryption using Key Agreement Algorithms Package

#	Objective	Threats
1	O.Hints_Enc_W	T.Clueless_With_En
2	O.Linkage_Enc_W	T.Assumed_Identity_With_En

#	Objective	Threats
3	O.Correct_KA	T.Garble_With_De

6.1.2.11 PKI Based Entity Authentication Package

The mapping of threats to objectives for the PKI Based Entity Authentication package is shown in Table 6.23. Text that further supports the mapping is provided following Table 6.23.

Table 6.23 – Mapping of Threats to Objectives for PKI Based Entity Authentication Package

#	Threat	Objectives
1	T.Assumed_Identity_Auth	O.Linkage, O.I&A_Remote, O.Limit_Actions_Auth_Remote
2	T.Replay_Entity	O.Single_Use_I&A

T.Assumed_Identity_Auth states that a user may assume the identity of another user to perform entity based authentication. This threat is addressed by:

- **O.Linkage**, which states that the TSF shall use the correct user public for authentication.
- **O.I&A_Remote**, which states that the TSF shall uniquely identify all remote entities, and shall authenticate the claimed identify before granting a remote entity access to the TOE facilities.
- **O.Limit_Actions_Auth_Remote**, which states that the TSF shall restrict the actions a remote entity may perform before the TSF verifies the identity of the remote entity.

T.Replay_Entity states that an unauthorized user may replay valid authentication data. This threat is addressed by:

- **O.Single_Use_I&A**, which states that the TSF shall use the I&A mechanism that requires unique authentication information for each I&A.

Table 6.24 maps objectives for the PKI Based Entity Authentication Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.24.

Table 6.24 – Mapping of Objectives to Threats for PKI Based Entity Authentication Package

#	Objective	Threats
1	O.Continuous_I&A	T.Hijack
2	O.I&A_Remote	T.Assumed_Identity_Auth
3	O.Limit_Actions_Auth_Remote	T.Assumed_Identity_Auth

#	Objective	Threats
4	O.Linkage	T.Assumed_Identity_Auth
5	O.Single_Use_I&A	T.Replay_Entity

6.1.2.12 OCSP Package Security Objectives Rationale

The mapping of threats to objectives for the OCSP package is shown in Table 6.25. Text that further supports the mapping is provided following Table 6.25.

Table 6.25 – Mapping of Threats to Objectives for the OCSP Package

#	Threat	Objectives
1	T.DOS_OSCP	O.User_Override_Fresh_OCSP
2	T.Replay_OCSP_Info	O.Fresh_OCSP_Info
3	T.Wrong_OCSP_Info	O.Accurate_OCSP_Info, O.Auth_OCSP_Info

T.DOS_OSCP 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_Fresh_OCSP**, which states that the TSF shall permit the user to override the freshness requirement for the OCSP response.

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

- **O.Fresh_OCSP_Info**, which states that the TSF accept only reasonably current OCSP response information.

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.

Table 6.26 maps objectives for the OCSP package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.26.

Table 6.26 – Mapping of Objectives to Threats for the OCSP Package

#	Objective	Threats
1	O.Accurate_OCSP_Info	T.Wrong_OCSP_Info

#	Objective	Threats
2	O.Auth_OCSP_Info	T.Wrong_OCSP_Info
3	O.Fresh_OCSP_Info	T.Replay_OCSP_Info
4	O.User_Override_Fresh_OCSP	T.DOS_OCSP

6.1.2.13 CRL Verification Package Security Objectives Rationale

The mapping of threats to objectives for the CRL Verification package is shown in Table 6.27. Text that further supports for the mapping is provided following Table 6.27.

Table 6.27 – Mapping of Threats to Objectives for CRL Verification Package

#	Threat	Objectives
1	T.DOS_CRL	O.User_Override_Fresh_CRL
2	T.Replay_Revoc_Info_CRL	O.Fresh_Rev_Info
3	T.Wrong_Revoc_Info_CRL	O.Accurate_Rev_Info, O.Auth_Rev_Info

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_Fresh_CRL**, which states that the TSF shall permit the user to override the freshness requirement for CRL.

T.Replay_Revoc_Info_CRL states that the user may accept old revocation information resulting in accepting currently revoked certificate. This threat is mapped to:

- **O.Fresh_Rev_Info**, which states that the TSF shall accept only reasonably current CRL..

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.

Table 6.28 maps objectives for the CRL Verification package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.28.

Table 6.28 – Mapping of Objectives to Threats for the CRL Verification Package

#	Objective	Threats
1	O.Accurate_Rev_Info	T.Wrong_Revoc_Info_CRL

#	Objective	Threats
2	O.Auth_Rev_Info	T.Wrong_Revoc_Info_CRL
3	O.Fresh_Rev_Info	T.Replay_Revoc_Info_CRL
4	O.User_Override_Fresh_CRL	T.DOS_CRL

6.1.2.14 Audit Management Package Security Objectives Rationale

The mapping of threats to objectives for the Audit Management package is shown in Table 6.29. Text that further supports for the mapping is provided following Table 6.29.

Table 6.29 – Mapping of Threats to Objectives for Audit Management Package

#	Threat	Objectives
1	T.Accountability	O.Audit_User
2	T.Audit_Excess	O.Audit_Select
3	T.Audit_Fill	O.Audit_Select
4	T.Audit_Modify	O.Audit_Protect
5	T.Audit_Unreadable	O.Audit_Readable
6	T.No_Audit	O.Audit

T.Accountability states that the security relevant audit events cannot be linked to individual actions. This threat is mapped to:

- **O.Audit_User**, which states that the TSF shall be capable of associating audit events with individual users.

T.Audit_Excess states that the security audit log has excessive data for analysis. This threat is mapped to:

- **O.Audit_Select**, which states that the TSF shall permit authorized users to select auditable events.

T.Audit_Fill states that the security audit log gets filled too fast to be of practical use. This threat is mapped to:

- **O.Audit_Select**, which states that the TSF shall permit authorized users to select auditable events.

T.Audit_Modify states that the accuracy of the security audit log cannot be trusted since unauthorized modification may have been made. This threat is mapped to:

- **O.Audit_Protect**, which states that the TSF shall protect the security audit log from unauthorized modifications.

T.Audit_Unreadable states that the audit log cannot be read and interpreted by human beings and hence security relevant events cannot be investigated. This threat is mapped to:

- **O.Audit_Readable**, which states that the TSF shall be able to generate human readable reports from the audit log.

T.No_Audit states that there is no audit log to investigate security relevant events. This threat is mapped to:

- **O.Audit**, which states that the TSF shall audit security relevant events.

Table 6.30 maps objectives for the Audit Management package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.30.

Table 6.30 – Mapping of Objectives to Threats for Audit Management Package

#	Objective	Threats
1	O.Audit	T.No_Audit
2	O.Audit_Protect	T.Audit_Modify
3	O.Audit_Readable	T.Audit_Unreadable
4	O.Audit_Select	T.Audit_Excess, T.Audit_Fill
5	O.Audit_User	T.Accountability

6.1.2.15 Continuous Authentication Package

The mapping of threats to objectives for the Continuous Authentication package is shown in Table 6.31. Text that further supports the mapping is provided following Table 6.32.

Table 6.31 – Mapping of Threats to Objectives for Continuous Authentication Package

#	Threat	Objectives
1	T.Hijack	O.Continuous_I&A

T.Hijack states that an unauthorized user may hijack an authenticated session. This threat is addressed by:

- **O.Continuous_I&A**, which states that the TSF shall continuously authenticate the entity.

Table 6.32 maps objectives for the Continuous Authentication Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.24.

Table 6.32 – Mapping of Objectives to Threats for Continuous Authentication Package

#	Objective	Threats
1	O.Continuous_I&A	T.Hijack

6.2 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.

6.2.1 Functional Security Requirements Rationale

The mapping of all security objectives to functional requirements (components) or to assumptions is provided in Table 6.33. Rationale for the base TOE security functional requirements mapping and for each package are described in separate subsections.

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

Table 6.33 – Security Objective to Functional Component Mapping

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

Table 6.33 (continued)

#	Objective	Functional Components
Mapping for Objectives for the Environment		
1	OE.Authorized_Users	Defined in the Administrator and User Guides under AGD_ADM.1 and AGD_USR.1, respectively
2	OE.Configuration	Defined in startup and installation guides under ADO_IGS.1
3	OE.Crypto	FCS_CRM_FPS.1
4	OE.Low	Defined in the SOF analysis and vulnerability assessment.
5	OE.Physical_Security	Defined as part of the physical security policy in AGD_ADM.1 and AGD_USR.1
6	OE.PKI_Info	FDP_ITC_PKI_INF.1
7	OE.Time	FPT_STM.1
Mapping for CPV – Basic Package		
1	O.Availability	FDP_DAU_CPV_CER.1
2	O.Correct_Time	FDP_DAU_CPV_INI.1
3	O.Current_Certificate	FDP_DAU_CPV_CER.1
3	O.Get_KeyInfo	FDP_DAU_CPV_OUT.1
5	O.Path_Find	FDP_CPD.1
6	O.Trusted_Keys	FDP_DAU_CPV_INI.1
7	O.User	FDP_DAU_CPV_CER.2
8	O.Verified_Certificate	FDP_DAU_CPV_CER.1
9	O.Valid_Certificate	FDP_DAU_CPV_CER.1
Mapping for CPV – Basic Policy Package		
1	O.Provide_Policy_Info	FDP_DAU_CPV_INI.2, FDP_DAU_CPV_OUT.2
Mapping for CPV – Policy Mapping Package		
1	O.Map_Policies	FDP_DAU_CPV_INI.3, FDP_DAU_CPV_CER.3, FDP_DAU_CPV_OUT.3
2	O.Policy_Enforce	FDP_DAU_CPV_INI.3, FDP_DAU_CPV_CER.3, FDP_DAU_CPV_OUT.3
Mapping for CPV – Name Constraints Package		
1	O.Authorised_Names	FDP_DAU_CPV_INI.4, FDP_DAU_CPV_CER.4, FDP_DAU_CPV_CER.5

Table 6.33 (continued)

#	Objective	Functional Components
Mapping for PKI Signature Generation Package		
1	O.Give_Sig_Hints	FDP_ETC_SIG.1
Mapping for PKI Signature Verification Package		
1	O.Use_Sig_Hints	FDP_ITC_SIG.1,
2	O.Linkage_Sig_Ver	FDP_DAU_SIG.1
Mapping for PKI Encryption using Key Transfer Algorithms Package		
1	O.Hints_Enc_WO	FDP_ETC_ENC.1
2	O.Linkage_Enc_WO	FDP_ETC_ENC.1, FDP_DAU_ENC.1
Mapping for PKI Encryption using Key Agreement Algorithms Package		
1	O.Hints_Enc_W	FDP_ETC_ENC.2
2	O.Linkage_Enc_W	FDP_ETC_ENC.2, FDP_DAU_ENC.2
Mapping for PKI Decryption using Key Transfer Algorithms Package		
2	O.Correct_KT	FDP_ITC_ENC.1
Mapping for PKI Decryption using Key Agreement Algorithms Package		
1	O.Hints_Dec_W	FDP_ITC_ENC.2
2	O.Linkage_Dec_W	FDP_DAU_ENC.3, FDP_ITC_ENC.2
3	O.Correct_KA	FDP_ITC_ENC.2
Mapping for PKI Based Entity Authentication Package		
1	O.I&A_Remote	FIA_UAU.1;1, FIA_UID.1;1
2	O.Limit_Actions_Auth_Remote	FIA_UAU.1;1, FIA_UID.1;1
3	O.Linkage	FIA_UAU_SIG.1
4	O.Single_Use_I&A	FIA_UAU.4
Mapping for Online Certificate Status Protocol Client Package		
1	O.Accurate_OCSP_Info	FDP_DAU_OCS.1
2	O.Auth_OCSP_Info	FDP_DAU_OCS.1
3	O.Fresh_OCSP_Info	FDP_DAU_OCS.1
4	O.User_Override_Fresh_OCSP	FDP_DAU_OCS.1

Table 6.33 (concluded)

Mapping for Certificate Revocation List (CRL) Validation Package		
1	O.Accurate_Rev_Info	FDP_DAU_CRL.1
2	O.Auth_Rev_Info	FDP_DAU_CRL.1
3	O.Fresh_Rev_Info	FDP_DAU_CRL.1
4	O.User_Override_Fresh_CRL	FDP_DAU_CRL.1
Mapping for Audit Management Package		
1	O.Audit	FAU_GEN.1
2	O.Audit_Protect	FAU_STG.1
3	O.Audit_Readable	FAU_SAR.1
4	O.Audit_Select	FAU_SEL.1
5	O.Audit_User	FAU_GEN.2
Mapping for Continuous Authentication Package		
1	O.Continuous_I&A	FIA_UAU.6:1, FIA_UAU.6:2

6.2.1.1 Security Objectives for the TOE Rationale

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

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

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

- FIA_ATD.1, User attribute definition, which requires that the TSF shall maintain the roles for individual users. This requirement ensures that all users are identified with a role or roles that provide certain permissions and access.
- FIA_UAU.1, Timing of authentication, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf

of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated.

- FIA_UID.1, Timing of identification, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified.

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

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

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

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

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

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

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

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

requirement restricts the number of consecutive unsuccessful authentication attempts.

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

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

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

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

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

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

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

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

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

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

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

- FMT_MTD.1, Management of TSF data, which requires that the TSF shall restrict the ability to perform operations specified by the ST author on TSF data specified by the ST author to roles specified by the ST author. This requirement ensures

that authorized users and their actions are defined for specified TSF data, including trust anchors.

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

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

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

6.2.1.2 Security Objectives for the Environment Rationale

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

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

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

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

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

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

number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules in the TOE shall be validated at FIPS 140 series Level 1. This environmental security objective is met by:

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

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

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

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

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

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

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

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

6.2.1.3 Certification Path Validation – Basic Package Rationale

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

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

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

- FDP_DAU_CPV_INI.1, Certification path initialisation – basic, which requires that the TSF obtain the current time called “current-time’ from a reliable source.

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

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

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

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

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

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

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

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

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

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

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

- FDP_DAU_CPV_CER.1, Certificate processing – basic, which requires that the TSF accept certificates only with verifiable signatures.

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

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

6.2.1.4 Certification Path Validation – Basic Policy Package Rationale

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

- FDP_DAU_CPV_INI.2, Certification path initialisation – basic policy, which requires that The TSF shall use the initial-certificate-policies provided by user roles specified by the ST author.
- FDP_DAU_CPV_OUT.2, Certification path output – basic policy, which requires that The TSF shall use output the certificate policies using the following rule:

intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies.

6.2.1.5 Certification Path Validation – Policy Mapping Package Rationale

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

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

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

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

6.2.1.6 Certification Path Validation – Name Constraints Package Rationale

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

- FDP_DAU_CPV_INI.4, Certification path initialisation – names, which requires that the TSF initialize the following: permitted-subtrees = ∞, excluded-subtrees = $\varnothing$
- FDP_DAU_CPV_CER.4, Intermediate certificate processing – name constraints, which requires that the TSF accept a certificate only if the conditions specified by the requirement, including verification of authorization, is satisfied.
- FDP_DAU_CPV_CER.5, Intermediate Certificate processing – name constraints, states that the TSF shall use the intermediate certificate to update the following states: permitted-subtrees and excluded-subtrees

6.2.1.7 PKI Signature Generation Package Rationale

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

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

6.2.1.8 PKI Signature Verification Package Rationale

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

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

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

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

6.2.1.9 PKI Encryption using Key Transfer Algorithms Package Rationale

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

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

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

- FDP_ETC_ENC.1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF use the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.
- FDP_DAU_ENC.1, PKI Encryption Verification – Key Transfer, which requires that the TSF apply verification checks for key transfer as selected by the ST author.

6.2.1.10 PKI Encryption using Key Agreement Algorithms Package Rationale

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

- FDP_ETC_ENC.2, Export of PKI Encryption – Key Agreement Algorithms, which requires that the TSF include the information with the encrypted data, such as the public key, as selected or assigned by the ST author and that the TSF use the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

O.Linkage_Enc_W states that the TSF shall use the correct user public key for key agreement during encryption. This objective is met by:

- FDP_ETC_ENC.2, Export of PKI Encryption – Key Agreement Algorithms, which requires that the TSF use the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.
- FDP_DAU_ENC.2, PKI Encryption Verification – Key Agreement, Subject, Decryptor, which requires that the TSF apply verification checks for key agreement as selected by the ST author.

6.2.1.11 PKI Decryption using Key Transfer Algorithms Package Rationale

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

- FDP_ITC_ENC.1, Import of PKI Encryption – Key Transfer Algorithms, which requires that the TSF use 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 and that the TSF will perform the decryption.

6.2.1.12 PKI Decryption using Key Agreement Algorithms Package Rationale

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

- FDP_ITC_ENC.2, Import of PKI Encryption – Key Agreement Algorithms, which requires that the TSF use the information from the encrypted data and information from Certification Path Validation to provide hints for selecting correct key agreement algorithm, certificates or keys.

O.Linkage_Dec_W states that the TSF shall use the correct user public key for key agreement during decryption. This objective is met by:

- FDP_ITC_ENC.2, Import of PKI Encryption – Key Agreement Algorithms, which requires that the TSF use the information from the encrypted data and information from Certification Path Validation to provide hints for selecting correct key agreement algorithm, certificates or keys.
- FDP_DAU_ENC.3, PKI Encryption Verification – Key Agreement, Subject, Encryptor, which requires that the TSF apply the following checks as selected by the ST author to verify the user public key using certification path validation.

O.Correct_KA states that the TSF shall use appropriate private key and key agreement algorithm. This objective is met by:

- FDP_ITC_ENC.2, Import of PKI Encryption – Key Agreement Algorithms, which requires that the TSF use the information from the encrypted data and information from Certification Path Validation to provide hints for selecting correct key agreement algorithm, certificates or keys.

6.2.1.13 PKI Based Entity Authentication Package Rationale

The PKI Based Entity Authentication package may or may not be included in an ST, depending on the functionality of the application. If this package is included, certain requirements, including FIA_UAU.1and FIA_UID.1, must be iterated by the ST author in order to differentiate between TOE users and users who are remote entities. The latter users, those who are remote entities, are the ones addressed by this package.

-

O.I&A_Remote states that the TSF shall uniquely identify all remote entities, and shall authenticate the claimed identity before granting a remote entity access to the TOE facilities. This objective is met by:

- FIA_UAU.1;1, Timing of authentication – remote entity, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated. The user in this case is the remote entity.
- FIA_UID.1;1, Timing of identification – remote entity, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified. The user in this case is the remote entity.

O.Limit_Actions_Auth_Remote states that the TSF shall restrict the actions a remote entity may perform before the TSF verifies the identity of the remote entity. This objective is met by:

- FIA_UAU.1;1, Timing of authentication – remote entity, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated. The user in this case is the remote entity.
- FIA_UID.1;1, Timing of identification – remote entity, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified. The user in this case is the remote entity.

O.Linkage states that the TSF shall use the correct user public key for authentication. This objective is met by:

- FIA_UAU_SIG.1, Entity authentication, which requires that the TSF use the following information from Certification Path Validation to verify the signature on signed data: subject public key algorithm, subject public key, subject public key parameters, and that the TSF perform additional checks as specified by the ST author.

O.Single_Use_I&A states that the TSF shall use the I&A mechanism that requires unique authentication information for each I&A. This objective is met by:

- FIA_UAU.4, Single-use authentication mechanisms, which requires that the TSF prevent reuse of authentication data.

6.2.1.14 Online Certificate Status Protocol Package Rationale

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

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

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

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

O.Fresh_OCSP_Info states that the TSF may accept only reasonably current OCSP responses. This objective is met by:

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

O.User_Override_Fresh_OCSP states that the TSF shall permit the user to override the freshness requirement for OCSP response. This objective is met by:

- FDP_DAU_OCS.1, Basic OCSP Client, which requires that a role or roles specified by the ST author be able to override the freshness requirement for revocation information.

6.2.1.15 Certificate Revocation List (CRL) Validation Package Rationale

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

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

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

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

O.Fresh_Rev_Info states that the TSF shall accept only reasonably current CRL. This objective is met by:

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

O.User_Override_Fresh_CRL states that the TSF shall permit the user to override the freshness requirement for CRL. This objective is met by:

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

6.2.1.16 Audit Management Package Rationale

O.Audit states that the TSF shall audit security relevant events. This objective is met by:

- FAU_GEN.1, Audit data generation, which requires that the TSF shall be able to generate an audit record of the specified auditable events and that the audit shall contain certain specified data.

O.Audit_Protect states that the TSF shall protect the security audit log from unauthorized modifications. This objective is met by:

- FAU_STG.1, Protected audit trail storage, which requires that the TSF shall protect the stored audit records from unauthorised deletion and modification as specified by the ST author.

O.Audit_Readable states that the TSF shall be able to generate human readable reports from the audit log. This objective is met by:

- FAU_SAR.1, Audit review, which requires that the TSF be able to generate reports from the audit log that are suitable for a user to read and be able to interpret information.

O.Audit_Select states that the TSF shall permit authorized users to select auditable events. This objective is met by:

- FAU_SEL.1, Selective audit, which requires that the TSF provide the capability for authorized users to select auditable events.

O.Audit_User states that the TSF shall be capable of associating audit events with individual users. This objective is met by:

- FAU_GEN.2, User identity association, which requires that the TSF be able to associate each auditable event with the identity of the user that caused the event.

6.2.1.17 Continuous Authentication Package Rationale

The Continuous Authentication package may or may not be included in an ST, depending on the functionality of the application. If this package is included,

requirement FIA_UAU.6 must be iterated by the ST author in order to differentiate between TOE users and users who are remote entities. The latter users, those who are remote entities, are the ones addressed by this package.

O.Continuous_I&A states that the TSF shall continuously authenticate the entity. This objective is met by:

- FIA_UAU.6:1, Re-authenticating remote entity, which requires that the TSF re-authenticate the remote entity under the conditions specified by the ST author.
- FIA_UAU.6:2, Re-authenticating user, which requires that the TSF re-authenticate the users other than the remote entity under the conditions specified by the ST author.

6.2.2 Assurance Requirement Rationale

This PP family includes a choice of EALs that are chosen by the PP/ST author. An EAL 3 with augmentation PP will be selected for TOEs that require a moderate level of independently assured security and require a thorough investigation of the TOE and its development without substantial re-engineering. EAL 3 provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation, and the high-level design of the TOE to understand the security behaviour. The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification and high-level design, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities. EAL 3 is augmented with ALC_FLR.1 to track and correct the reported and found security flaws in the product.

An EAL 4 with augmentation PP will be selected for TOEs that require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs. 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 behaviour. Assurance is additionally gained through an informal model of the TOE security policy. EAL 4 represents a meaningful increase in assurance from EAL 3 by requiring more design description, a subset of the implementation, and improved mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development or delivery. EAL 4 is augmented with ALC_FLR.1 to track and correct the reported and found security flaws in the product.

6.2.3 Strength of Function Rationale

The TOE is assumed to be designed to protect against “low” attack potential. Thus, based on the CEM Annex B, Table B.2, the minimum strength of function is SOF Basic. The strength of cryptographic algorithms is outside the scope of the CC. Strength of function only applies to non-cryptographic, probabilistic or permutational mechanisms. The SOF requirement applies to the identification and authentication functionality for the TOE. Note that the I&A mechanism used in the TOE is application-specific and SOF

analysis must be performed as part of ST development. PPs in this family require a SOF rating of SOF Basic or higher.

A SOF rating reflects the attacker, described in terms of attack potential, against which the probabilistic or permutational security function is designed to protect. To determine a SOF rating for the I&A functionality provided in the TOE, the developer of the ST must calculate the attack potential. One way to calculate the attack potential is to use Table B.3 from the CEM Annex B to calculate a numerical score for attack potential and then use Table B.4 from the CEM Annex B to translate the number into a qualitative attack potential and an SOF rating. For example, using Table B.3, assuming a layman with no knowledge of the TOE and no equipment, with > 1 month elapsed time, and > 1 month access to the TOE results in a score of 17 for attack potential. Note that a brute force attack on the I&A mechanism is obvious and hence the corresponding identifying values are all zero.

Using Table B.4 (duplicated below), the resistance to attack with attack potential score translates to an attack potential of “low”. Again, using Table B.2 or B.4, a SOF rating of SOF Basic is required for attack potential of “low”.

Table B.4 from CEM Annex B

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

6.3 Dependency Rationale

Table 6.34 – Functional Requirements Dependencies

#	Requirement	Dependencies
Base Functional Requirements		
1	FDP_ACC.1	FDP_ACF.1
2	FDP_ACF.1	FDP_ACC.1, FMT_MSA.3
3	FIA_AFL.1	FIA_UAU.1
4	FIA_ATD.1	None
5	FIA_UAU.1	FIA_UID.1
6	FIA_UAU.7	FIA_UAU.1
7	FIA_UID.1	None
8	FMT_MSA.1	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1
9	FMT_MSA.3	FMT_MSA.1, FMT_SMR.1
10	FMT_MTD.1	FMT_SMF.1, FMT_SMR.1
11	FMT_SMF.1	None
12	FMT_SMR.2	FIA_UID.1
13	FPT_RVM.1	None
14	FPT_SEP.1	None
IT Environment Functional Requirements		
1	FCS_CRM_FPS.1	None
2	FDP_ITC_PKI_INF.1	None
3	FPT_STM.1	None
PKI Based Entity Authentication Package		
1	FIA_UAU.1 (includes iteration)	FIA_UID.1
2	FIA_UAU.4	None
3	FIA_UAU_SIG.1	FCS_COP.1 (see Note 4), FDP_DAU_CPV_OUT.1 (see Note 5)
4	FIA_UID.1 (includes iteration)	None
Continuous Authentication Package		
1	FIA_UAU.6	None

Table 6.34 (continued)

#	Requirement	Dependencies
	Audit Management Package	
1	FAU_GEN.1	FPT_STM.1 (See Note 1)
2	FAU_GEN.2	FAU_GEN.1, FIA_UID.1 (See Note 2)
3	FAU_SAR.1	FAU_GEN.1
4	FAU_SEL.1	FAU_GEN.1, FMT_MTD.1 (See Note 3)
5	FAU_STG.1	FAU_GEN.1
	CPV – Basic Package	
1	FDP_CPD.1	None
2	FDP_DAU_CPV_INI.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)
3	FDP_DAU_CPV_CER.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)
4	FDP_DAU_CPV_CER.2	FDP_DAU_CPV_CER.1
5	FDP_DAU_CPV_OUT.1	None
	CPV – Basic Policy Package	
1	FDP_DAU_CPV_INI.2	FDP_DAU_CPV_INI.1 (See Note 5)
2	FDP_DAU_CPV_OUT.2	FDP_DAU_CPV_OUT.1 (See Note 5)
	CPV – Policy Mapping Package	
1	FDP_DAU_CPV_INI.3	FDP_DAU_CPV_INI.2 (See Note 6)
2	FDP_DAU_CPV_CER.3	FDP_DAU_CPV_CER.2 (See Note 7)
3	FDP_DAU_CPV_OUT.3	FDP_DAU_CPV_OUT.2 (See Note 6)
	CPV – Name Constraints Package	
1	FDP_DAU_CPV_INI.4	FDP_DAU_CPV_INI.1 (See Note 5)
2	FDP_DAU_CPV_CER.4	FDP_DAU_CPV_CER.1 (See Note 5)
3	FDP_DAU_CPV_CER.5	FDP_DAU_CPV_CER.2 (See Note 5)
	PKI Signature Generation Package	
1	FDP_ETC_SIG.1	FCS_COP.1 (See Note 4)
	PKI Signature Verification Package	
1	FDP_ITC_SIG.1	None
2	FDP_DAU_SIG.1	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)

Table 6.34 (concluded)

#	Requirement	Dependencies
PKI Encryption using Key Transfer Algorithms Package		
1	FDP_ETC_ENC.1	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)
2	FDP_DAU_ENC.1	FDP_DAU_CPV_OUT.1 (See Note 5)
PKI Encryption using Key Agreement Algorithms Package		
1	FDP_ETC_ENC.2	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)
2	FDP_DAU_ENC.2	FDP_DAU_CPV_OUT.1 (See Note 5)
PKI Decryption using Key Transfer Algorithms Package		
1	FDP_ITC_ENC.1	FCS_COP.1 (See Note 4)
PKI Decryption using Key Agreement Algorithms Package		
1	FDP_ITC_ENC.2	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)
2	FDP_DAU_ENC.3	FDP_DAU_CPV_OUT.1 (See Note 5)
Online Certificate Status Protocol Client Package		
1	FDP_DAU_OCS.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)
Certificate Revocation List (CRL) Validation Package		
1	FDP_DAU_CRL.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)

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

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

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

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

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

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

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

References

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

Glossary of Terms

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

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

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

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

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

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

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

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

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

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

Path Processing
The means employed by a relying party to ensure that the certificates in a path leading from a relying party trust point to subscriber's public key certificate, are all valid. The validation activity includes chaining the subscriber and issuer names, using the subject public key from the parent

certificate to verify the signature on a certificate, applying constraints imposed by the various extensions in the certificate, verifying that none of the certificates have expired or been revoked, and other X.509 certification path validation rules.

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

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

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

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

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

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

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

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

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

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

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

Symmetric Key
A key that is used to both encrypt and decrypt information. Parties involved in using the key must keep the key secret; anyone with knowledge of the key could either originate or view encrypted information.

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

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

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

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

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

List of Acronyms

CA	Certification Authority
CAC	Common Access Card
CC	Common Criteria
CEM	Common Evaluation Methodology
CPV	Certification Path Validation
CRL	Certificate Revocation List
CRLDP	CRL Distribution Point
DH	Diffie Hellman
DISA	Defense Information Systems Agency
DN	Distinguished Name
DoD	Department of Defense
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECDH	Elliptic Curve Diffie Hellman
EFS	Encrypted File System
EKU	Extended Key Usage
FIPS	Federal Information Processing Standard
GMT	Greenwich Mean Time
HMAC	Hash based Message Authentication Code
IDP	Issuing Distribution Point
IEC	International Electrotechnical Committee
IETF	Internet Engineering Task Force
ISO	International Organisation for Standards
IT	Information Technology

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