# US Government

# Directory Protection Profile

# For

# Medium Robustness Environments



**Information Assurance Directorate**

July 25, 2007
Version 1.1

# 1 PREFACE

**Protection Profile Title:**

U.S. Government Protection Profile Directory for Medium Robustness Environments

**Criteria Version:**

This Protection Profile "*US Government Protection Profile Directory for Medium Robustness Environments"* (PP) was updated using Version 3.1 of the Common Criteria (CC).

Editor's note:  The purpose of this update was to bring the PP up to the new CC 3.1 standard without changing the authors' original meaning or purpose of the documented requirements.  The original PP was developed using version 2.x of the CC.  The CC version 2.3 was the final version 2 update that included all international interpretations.  CC version 3.1 used the final CC version 2.3 Security Functional Requirements (SFR)s as the new set of SFRs for version 3.1. Some minor changes were made to the SFRs in version 3.1, including moving a few SFRs to Security Assurance Requirements (SAR)s.  There may be other minor differences between some SFRs in the version 2.3 PP and the new version 3.1 SFRs.  These minor differences were not modified to ensure the author's original intent was preserved.

The version 3.1 SARs were rewritten by the common criteria international community.  The NIAP/CCEVS staff developed an assurance equivalence mapping between the version 2.3 and 3.1 SARs.  The assurance equivalent version 3.1 SARs replaced the version 2.3 SARs in the PP.

Any issue that may arise when claiming compliance with this PP can be resolved using the observation report (OR) and observation decision (OD) process.

Further information, including the status and updates of this protection profile can be found on the CCEVS website:  http://www.niap-ccevs.org/cc-scheme/pp/.  Comments on this document should be directed to ppcomments@missi.ncsc.mil.  The email should include the title of the document, the page, the section number, the paragraph number, and the detailed comment and recommendation.

## 2 RECORD OF RELEASE

[**(Back to TOC)**](#)

| Release # | Date | Area Affected | Comment |
|---|---|---|---|
| Release 1.0 | September 1, 2004 | Complete Document | Release of the NIAP evaluated PP |
| Release 1.1 | July 25, 2007 | Assurance requirements<br><br>Functional requirements<br>NIAP Interps<br>Rationale | Updated to CC version 3.1 |

# 3   FORWARD

**Revisions**

This document replaces previous versions of the 'Department of Defense Class 4 PKI Directory Protection Profile', and includes modifications for the following purposes:

to be consistent with the 'Protection Profile Consistency Guidance', 23 July 2002 for medium robustness, and guidance from the PP Review Board, and new Medium Robustness;

to include new Medium Robustness Assurance requirements;

to incorporate edits based on further review and comments received.

**Acknowledgements**

The author would like to thank the following individuals for their substantive contributions to the development of this PP:  Matt Hirsch, A&N Associates; Alethea Brown, A&N Associates.

**TABLE OF CONTENTS**

## TABLE OF FIGURES

## TABLE OF TABLES

{This page intentionally left blank}

# 1 INTRODUCTION

This Directory Protection Profile (PP) for Medium Robustness Environments is sponsored by the National Security Agency (NSA) to provide secure directory information services for Department of Defense (DoD) Systems, and is intended for the following uses:

For vendors and security evaluators, this PP defines the requirements that must be addressed by specific products as documented in vendor Security Targets (STs).

For system integrators, this PP is useful in identifying areas that need to be addressed to provide secure system solutions.

## 1.1 IDENTIFICATION

**Title:** U.S. Department of Defense (DoD) Directory Protection Profile (PP) for Medium Robustness Environments

**Sponsor:** National Security Agency (NSA)

**CC Version:** Common Criteria (CC) Version 3.1, and applicable interpretations.

**Registration:** <to be provided upon registration>

**Protection Profile Version:** Version 1.1, dated 25 July 2007.

**Evaluation Assurance Level:** U.S. DoD Medium Robustness Assurance consisting of: ADV_ARC.1, ADV_FSP.5, ADV_IMP.1, ADV_INT.3, ADV_TDS.4, AGD_OPE.1, AGD_PRE.1, ALC.CMC.4, ALC_CMS.4, ALC_DEL.1, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1, ALC_TAT.1, ATE_COV.2, ATE_DPT.3, ATE_FUN.1, ATE_IND.2, AVA_CCA_(EXT).1, AVA_VAN.4

**Keywords:** Directory, Repository, Replication, Chaining, Distributed Authentication, Medium Robustness Environments, LDAP, X.500, X.509, Public Key Infrastructure (PKI), Global Directory Service (GDS), Key Management Infrastructure (KMI), Department of Defense (DoD), Directory System Agent (DSA), Administrative Directory User agent (ADUA).

## 1.2 PROTECTION PROFILE OVERVIEW

This PP specifies the minimum-security requirements for directories (i.e., the Target of Evaluation (TOE)) used by the Department of Defense (DoD) in Medium Robustness Environments. The directory provides controlled access to a repository of information (RI) for a single classification or marking, and is considered sufficient protection for environments where the likelihood of an attempted compromise is medium. The target robustness level of "medium" is specified in the Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG) [2] and is further discussed in Section 3.0 of this PP. STs claiming compliance may consist of one or more devices, and, as a medium robustness TOE, must define its TOE to include all the components necessary to meet the security functional requirements, including the hardware.

The PP defines the requirements for a general-purpose directory that may be used in a variety of applications and systems, including Public Key Infrastructures (PKIs). The TOE for the directory

includes security requirements for identification and authentication (I&A), access control, non-repudiation, audit, trusted channel/path, and TSF management, self-protection, and data availability. A cryptographic module is required for the security mechanisms that use encryption and digital signatures, e.g., trusted channel and I&A, respectively.

STs that claim conformance to this PP shall meet a minimum standard of demonstrable-PP conformance as defined in section D3 of part 1.

Relative to these requirements the PP includes:

- assumptions about the security aspects of the environment in which the TOE will be used;

- threats that are to be addressed by the TOE;

- security objectives of the TOE and its environment;

- functional and assurance requirements to meet those security objectives; and

- rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

## 1.3   THE TOE AS A COMPONENT OF A SYSTEM

The PP includes security requirements associated with a directory server as part of a distributed directory system and as part of a larger system, e.g., a PKI.  As a component of these systems the TOE must work in concert with other components to provide system security services.  While the PP includes requirements for component security functions to support system security services, it doesn't specify 'how' the requirement must be met.  Therefore it does not specify protocols or standards for compliance.

However, when using the TOE as a system component, users may have protocol or standard compliance requirements and must understand the mechanisms used to comprise a system security function. To assist with this composition process compliant STs may  specify in their TOE Summary Specification Section the mechanisms, protocols, or standards used to meet these component requirements, e.g., replication and distributed I&A mechanisms.

### 1.3.1   The TOE As A Component of a Distributed Directory System

In this PP a distributed directory system is a directory service that resides on more than one directory server.  It may partition the repository information among the different servers and it may replicate the repository information among the different servers.  It may also comprise any combination of the following characteristics:

- runs homogeneous or heterogeneous directory server products;

- operates under single or multiple administrative management control(s);

- implements a single or multiple security policies;

- operates under a single or multiple organizational control(s).

The associated security requirements include a replication mechanism that ensures all associated security attributes are included with the replica data, and requirements for a distributed authentication mechanism. There is also a stated assumption that the directory server components in a distributed directory system have established trust that the access control, and identification and authentication security policies are understood and enforced.

### 1.3.2 The TOE As A Component of a Larger System or Infrastructure

A larger system may include a directory as its component, and it may have system-level security requirements that must be supported by its component directory, e.g., system-wide audit data analysis.

The associated security requirements for the directory as a general larger system component include requirements for system-wide audit data analysis, time synchronization, and availability of the directory information for other system component's security functions.

## 1.4 COMMON CRITERIA CONFORMANCE

### 1.4.1 PP Conformance Claim

This Protection Profile is Common Criteria Part 2 version 3.1 and Common Criteria Part 3 version 3.1, with U.S. DoD Medium Robustness Assurance. This PP is also conformant with CEM Supplement: ALC_FLR – Flaw Remediation.

### 1.4.2 STs Claiming Conformance to this PP

An ST claiming conformance to this PP must define its TOE to include all SFRs specified in Section 5.1 without reliance to its environment, and all components required for operation, including hardware components.

## 1.5 PROTECTION PROFILE CONTENTS AND ORGANIZATION

**Section 1** introduces this PP document through an overview, a statement of Common Criteria Conformance, and a description of this PP organization.

**Section 2** describes the TOE and the environment. This section also provides an overview of the security functionality provided upon conformance with this PP.

**Section 3** provides informative introductory text to help the reader gain an understanding of the various robustness levels and more importantly how to determine the proper robustness level for a given system. Additionally, Section 3 discusses the characteristics of environments and threat levels

appropriate for the TOE and specifies the TOE assumptions, threats, and organizational security policies.

**Section 4** identifies the security objectives satisfied by the TOE and the TOE environment.

**Section 5** specifies the functional and assurance requirements for the TOE and its IT environment.

**Section 6** provides the rationale for the security objectives and the security requirements. The objectives rationale shows that the security objectives address the assumptions, threats and policies. The requirements rationale shows that the requirements meet the objectives and that all dependencies are satisfied. In addition, rationale is provided for Assurance requirements.

**Section 7** contains expansions of acronyms used throughout this PP.

**Section 8** contains the references.

**Section 9** provides a glossary of terms.

# 2   TOE DESCRIPTION

## 2.1   PRODUCT TYPE

TOEs claiming conformance to this Protection Profile are directories that provide controlled access to a repository of information (RI) requiring protection at a Medium Robustness Level of Assurance at a single classification or marking.  The PP defines the security requirements for a general-purpose directory that may be used in a variety of mission critical applications and systems, including PKIs. For example, in a PKI the directory must ensure certificates and revocation lists are available for relying parties to use certificate-based security mechanisms (e.g., digital signature verification), and it must control access to this security data, e.g., only an authorized Certificate Authority (CA) can update a certain Certificate Revocation List (CRL) entry.

This PP defines the requirements for a directory which may or may not be a single directory server, but which must be able to function as part of a distributed directory system and as a component of an application system, e.g., PKI.  As described in Section 1.3.1, a distributed directory system comprises multiple individual directory servers that interoperate to form an overall distributed directory. Replication and authentication security requirements are included to support this.  As a component in a system, e.g., a PKI, the directory must support system-wide security services.  This includes controlled access to audit data for system-wide audit data analysis, and mechanisms to synchronize the directory's time with other system components.

Specific directory protocols and standards are not specified in the PP, and are only used to provide examples.  Interoperability issues and evaluation is outside of its scope.  However, as a directory capable of operating within a distributed directory system and as a component in a system, a TOE claiming conformance to this PP are requested to specify in their ST the mechanisms they use and the interfaces available for functions that require interoperability beyond the scope of the TOE, e.g., replication, distributed authentication, trusted channel.

Directories can be implemented in various ways and may use several different components and technologies as part of a system.  Some of these components have existing PP's, e.g., Certificate Issuing and Management Components (CIMC), and there are also PP's for technologies that may be used to implement a Directory system, e.g., a web server.   The PP's that may be applicable for a system implementation are listed in Section 8.3, Related Protection Profiles.  Users that want an evaluation of a directory that includes these other components are to specify that they expect all applicable PPs to be compliant in their acquisition request.

The TOE functional security requirements, i.e., security services, can be categorized as follows, and are described in Section 2.4:

- Access Control,

- Identification and Authentication,

- Replication,

- Non-repudiation,

- Audit,

- Trusted Channel/Path,

- Cryptographic Support,

- Administration,

- Internal Capabilities

The following provides more information on the components of the TOE, its users, and the security services.

## 2.2  TOE BOUNDARY

The TOE boundary, illustrated in Figure 2.1 below, includes all hardware and software components necessary to provide secure directory service.  The TOE includes functionality required to administer and manage the directory both locally and remotely.   The trusted local terminal interface (i.e., local console) is included in the TOE.  The interface for trusted remote access is not included in the TOE to enable applications to use interfaces appropriate for their system architecture.  The TOE does require the remote trusted interfaces establish a trusted channel with the TOE and a trusted path with its users, and that the users authenticate to the TOE.

While this document does not dictate the required components, Figure 2.2 provides an example implementation that includes:

- A Directory Service application, e.g., DSA;

- A Directory Information Base, i.e., the repository information;

- Administrative functionality, e.g., ADUA;

- A Cryptographic Module;

- An Operational Platform that provides data storage, network  interface and includes an operating system, a hardware platform, and local console.

Figure 2.3 provides an illustration of the TOE as a component in a distributed directory system and the security functions directly related to distributed operations, i.e., replication, distributed authentication, access control for PKI components, and non-repudiation.

**Figure 2.1 – Directory TOE and Users**
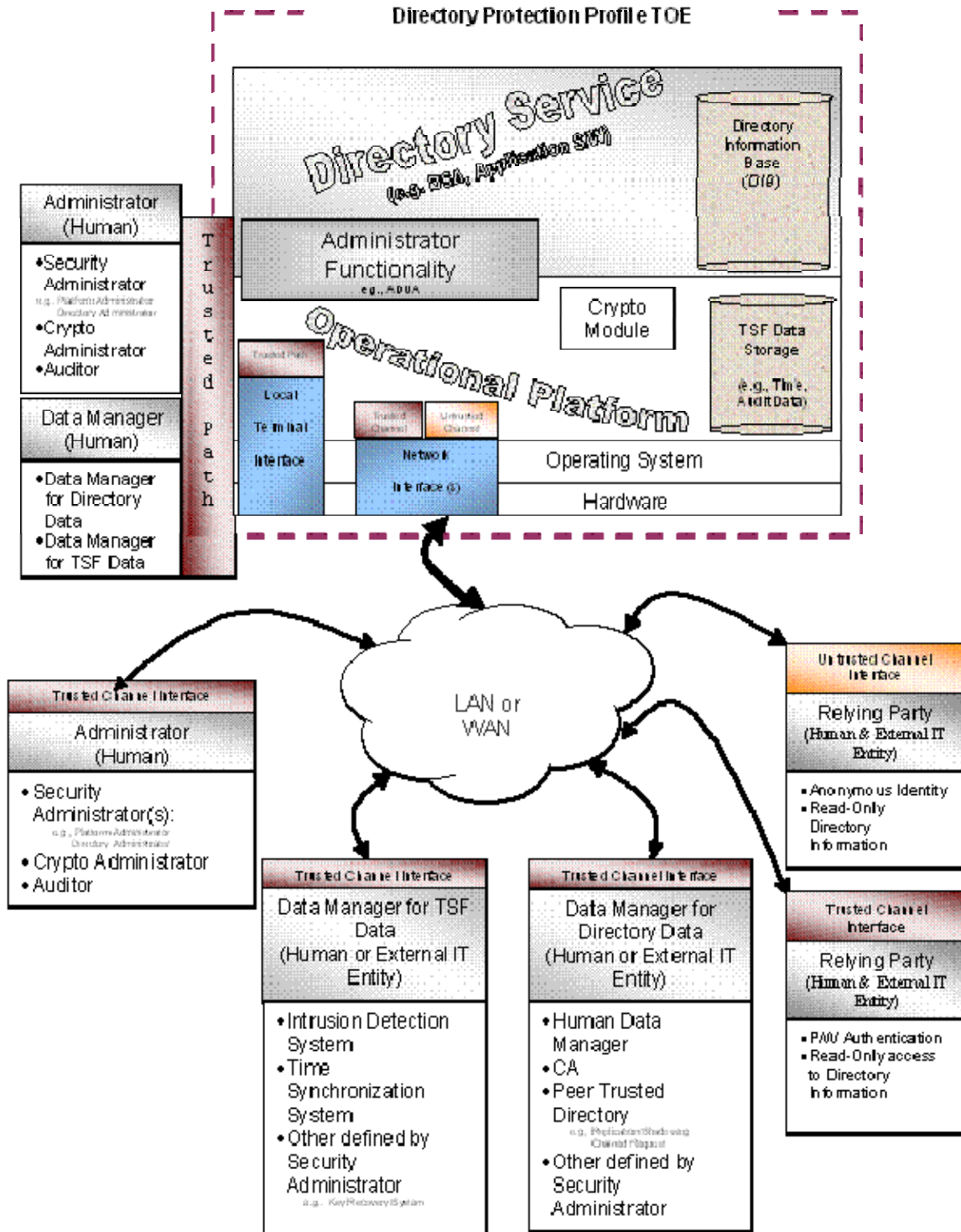
**Figure 2.2 – Directory Security Services**
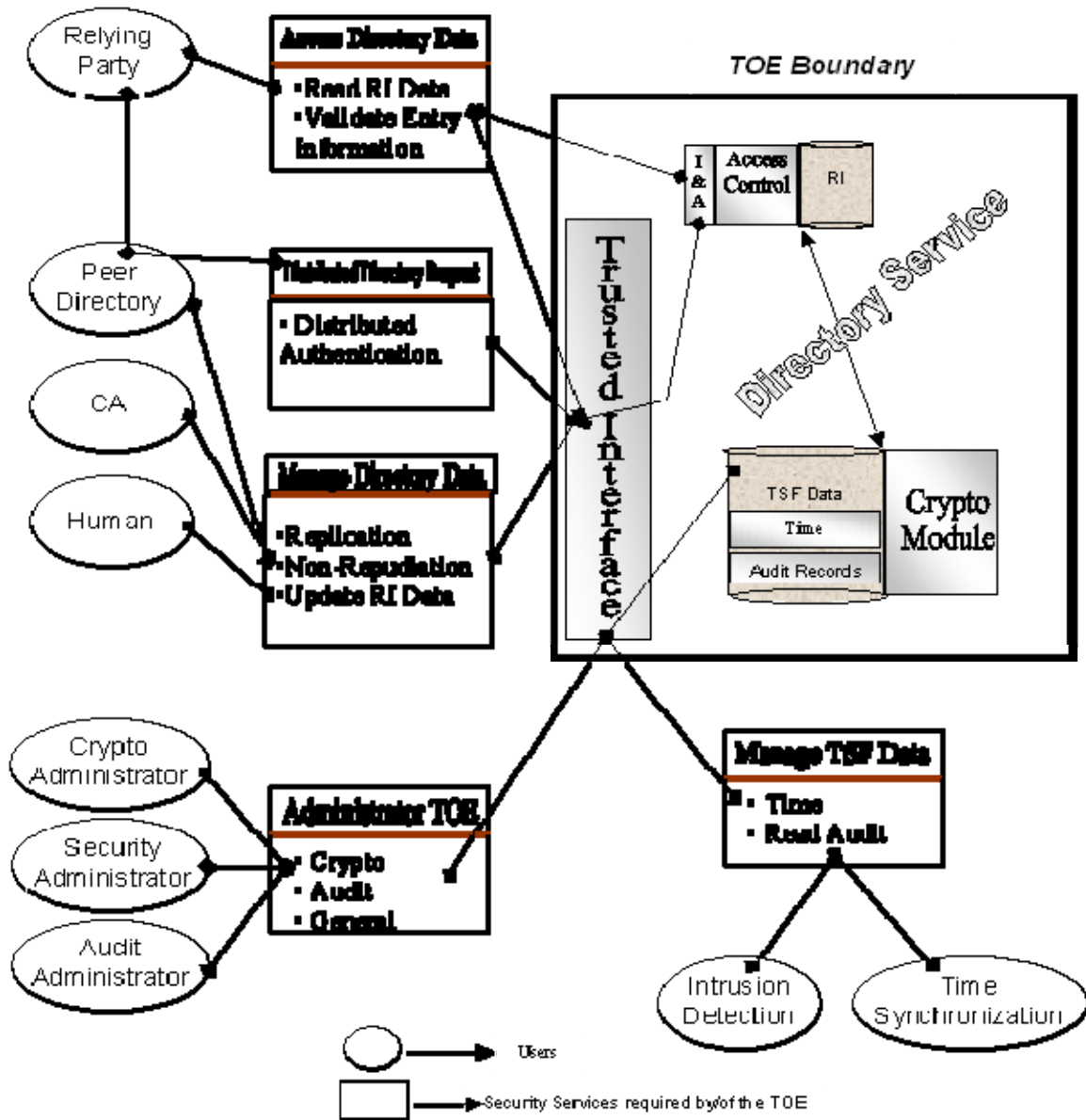
**Figure 2.3 – TOE in a Distributed Directory**
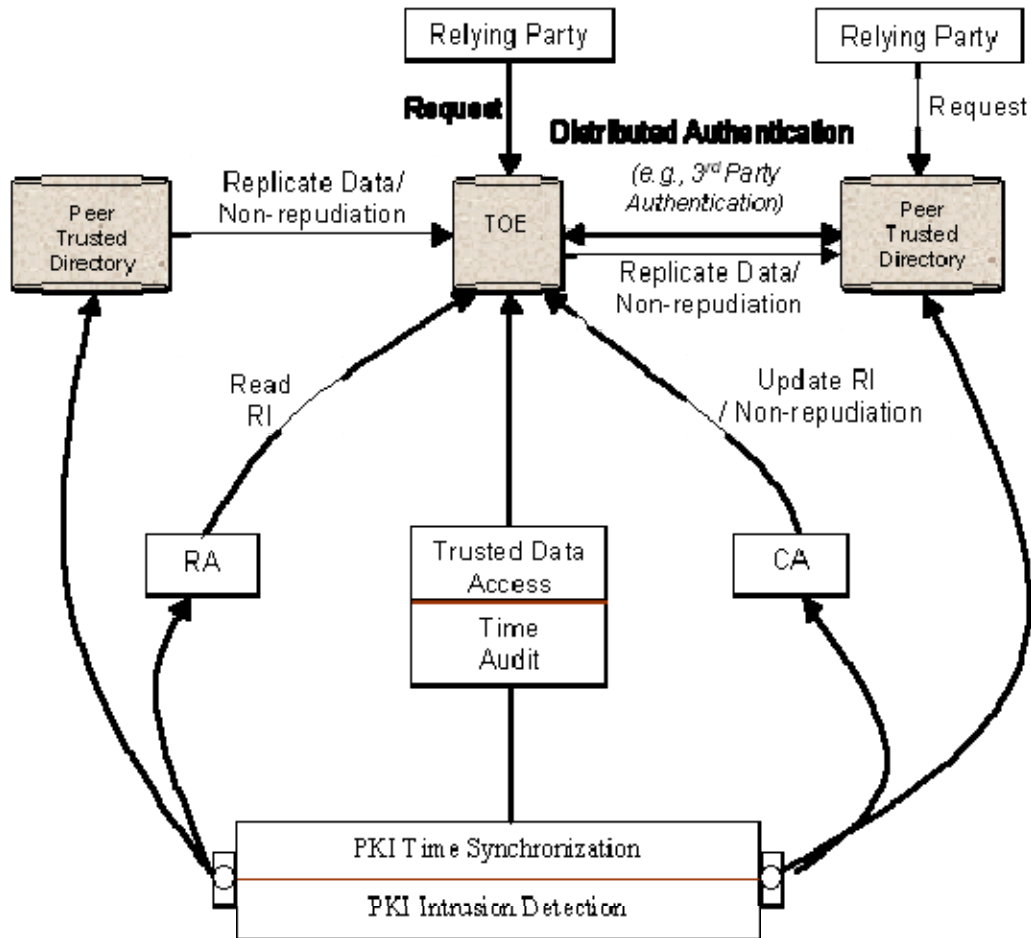
## *2.3 USERS*

As illustrated in Figure 2.2, this PP defines three kinds of users:  Relying Parties, Administrators, and Data Managers.  The following describes how these users access the TOE, the security services they access, and how they are represented as roles in the TOE.

**RELYING PARTIES** are untrusted human users or external IT entities that rely on the repository of information maintained by the directory.  This PP defines requirements that ensure these users only have read access to the repository information, and their identity may be authenticated using a certificate or a password, or they may be anonymous.  All access to the TOE is remote and may be over either a trusted channel, required for password authentication, or an untrusted channel.   The TOE requires a single role, Relying Parties, to support these users.

**ADMINISTRATORS** are trusted human users who are responsible for the management and operation of the TOE.  They may access the TOE locally over a trusted path or remotely over a crypto-based trusted channel.  Remote administrators must be authenticated with a certificate; local administrators may be authenticated using a password.

Administrators have expertise in aspects of operating the TOE and are responsible for its hardware, software and security functions.  To isolate administrative actions, the PP requires at least three administrative roles, a crypto administrator for the cryptographic functions, an auditor for the audit functions, and a general security administrator for general administrative responsibilities.  It's anticipated that a compliant implementation may refine and iterate the Security Administrator role as necessary to support component parts of the TOE, e.g., a Directory Administrator and a Platform Administrator.  The Security Administrator(s) grant specific Data Managers access to a set of trusted data.

**DATA MANAGERS** are trusted human users or external IT entities responsible for providing or accessing a set of trusted data (TSF data).  These managers are the authoritative source for the data provided by the directory service or used by the TOE, or they may need access to trusted data.  Examples of data managers include:

CA's that provide certificates and Revocation Lists (RLs),

human data managers that update entries in the directory as granted access by an administrator, peer trusted directories that update or receive the repository information through a replication process, and support a distributed authentication mechanism, and

external trusted entities that update time, and an intrusion detection system that reads audit records.

The TOE requires a single role, Data Manager, to support these users. The PP requires the Data Manager role has a user identity associated with a security administrator-specified set of trusted data for which they have access. For example, a CA (user) with identity CA_1 has update access to a set of repository entries. This role is defined in this manner to support various architectures and policies regarding access and maintenance of the trusted data in the TOE. The ST author may refine the Data Manager role and its assignments to reflect the implementation.

## 2.4   SECURITY SERVICES

The TOE functional security requirements can be categorized as follows:

• Access control,

• Identification and Authentication,

• Replication,

• Non-repudiation,

• Audit,

• Trusted Channel/Path,

• Cryptographic Support,

• Administration and Management,

• Internal Capabilities.

**Access Control:  t**he TOE includes an access control security policy that restricts access to the directory information.   Relying Parties only have read-only access, and only security administrator-specified trusted data managers have update access.

The access control decisions are based on the security attributes for the objects that constitute the repository information, and the subject attributes of the requesters. The object attributes are in the form of ACI items, and the subject attributes include distinguished name, user group, role, and authentication level. The ACI item attribute associates protected items and user classes with permissions. Rather than each object having its own ACI item (or set of permissions), the directory has a set of ACI items for all the repository data. Each ACI item grants or denies permissions in regard to a set of specified users and protected items. The scope of the protected items can be a single entry, attribute, or subtree of entries, resulting in an access control decision for a single request being based on multiple ACIs. Other ACI attributes include priority, and required authentication level.

**Identification and Authentication:** the TOE requires multiple Identification and Authentication (I&A) mechanisms for access to services residing on the TOE. The type of authentication mechanism required depends on the type of user, their credentials, and their location. Local administrators and data managers may authenticate using a password. Remote access for these users requires certificate-based authentication, and the access must be over a trusted channel.

The TOE requires several authentication options for Relying Parties. Anonymous access by Relying Parties is permitted and the TOE assigns the identity 'anonymous' for these users, and the communication may be over an untrusted channel. This identity is used for access control decisions. All non-anonymous authentication for relying parties must be over a trusted channel. Relying Parties may authenticate using a password, a certificate, or a distributed authentication mechanism that is commensurate in strength to the other required relying party authentication mechanisms.

A distributed authentication mechanism is one that supports a distributed directory. It may allow the authentication data, the I&A mechanism, and the repository information being accessed to reside on separate servers. Two examples of distributed authentication mechanisms that a compliant TOE may implement are '3rd party introduction' and '3rd party presentation'. '3rd Party introduction' trusts that the peer directory correctly verified the authentication credentials of the relying party before passing the chained request to the TOE. '3rd Party presentation' trusts that the peer directory ensured the integrity and, if necessary, the confidentiality of the authentication credentials passed to the TOE as part of the chained request. Both these mechanisms require that trust is established with the peer directory.

**Replication:** the TOE includes requirements to support directory replication. Directory replication is the process used in a distributed directory environment in which a replica of a portion of the repository information is copied to and/or from other directories. This increases the availability of the Directory's repository data within a system. The TOE requires the TSF to ensure the integrity of the replicated data it receives or sends and to ensure the security attributes are associated with the data.

**Non-repudiation:** the TOE requires non-repudiation services to support the TOE's role in a PKI to make RLs and certificates available according to their certificate policies. The non-repudiation service applies to the transmission of repository data to or from the TOE through either updates to the data from a data manager or replication among peer trusted directories. The non-repudiation requirements include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that evidence of receipt the TOE is waiting for is overdue.

**Audit:** the audit requirements for the TOE include generating records for auditable events, alarms and audit management. To isolate administrative actions the TOE requires that only the auditor role view, search, and sort the audit trail. Only the security administrator configures the behavior of the audit mechanisms including, setting thresholds, configuring auditable events, backs-up and deletes audit data, and manages audit data storage.

The TOE requires a minimum set of auditable events, and the minimum contents of the audit records. TOEs claiming compliance to this PP may include additional auditable events and record contents. If they also include additional functional requirements audit records must able to be generated for the associated security relevant events.

In addition to generating auditable events, the TOE must monitor their occurrences and provide a Security Administrator-configurable threshold for determining a potential security violation. Once the TOE has detected a potential security violation, an alarm is generated and a message is displayed at the TOE's local console as well as each active remote auditor and security administrator active sessions and those initiated before the alarm has been acknowledged. The message must contain the potential security violation and the TOE must make accessible all audit records associated with the potential security violation. The message will continue to be displayed until it has been acknowledged.

**Trusted Channel/Path:** the TOE is required to provide two types of encrypted communications: trusted channel and trusted path. Trusted channel refers to the encrypted connection that prevents disclosure and detects modification of data transmitted between the TOE and an external IT entity, e.g., an encrypted connection between the TOE and a trusted peer directory. Trusted path refers to the encrypted connection that prevents disclosure and detects modification of data transmitted between a human user and the TOE, e.g., a remote administration.

The trusted channel must be used for all password-based authentication functions, replication operations, and remote management of the directory service data. While the external trusted IT entities may initiate communications, it may be the case that the TOE is required to perform a "pull" operation (e.g., obtaining time from a time server).

The trusted path must be used for relying party password-based authentication and all remote administration actions.

**Cryptographic Support:** the TOE includes security functions that depend on cryptographic operations. These include:

digital signature verification for authentication;

• encryption to prevent disclosure for a trusted channel, and a trusted path;

• cryptographic function to ensure integrity for self testing stored TSF data and TSF executable code, a trusted channel, and a trusted path;

• random number generation and a hashing function to support the above operations.

For medium robustness, a symmetric key size of at least 128 bits is required. For Digital Signatures, an equivalent degree of "security" is required for key cryptographic parameters in the algorithms used. For both the DSA and RSA algorithms, modulus sizes of at least 2048 are required to provide this degree of security. For medium robustness it's also required that applicable cryptographic functionality be FIPS 140-2 validated.

The TOE requires the following algorithms be implemented by a cryptographic module:

• Encryption/Decryption using AES

• Digital Signature Generation/Verification using rDSA or ECDSA. Note: the DSA algorithm described in the DSS (FIPS 186-2) is limited to a maximum modulus size of 1024 bits and is therefore not suitable for implementing digital signature functionality for medium robustness.

To support these operations the TSF must provide the following cryptographic key management functions:

• Key generation,

• Key establishment using: key agreement, key transport, manual loading, or automated loading; and

• Key destruction.

**Administration and Management:** the TOE includes functions and roles for administration and management of the trusted data. As described above in Section 2.3, the TOE includes three separate administrative roles, Cryptographic Administrator, Auditor, and Security Administrator, and a single trusted Data Manager role. These roles may be refined as necessary to support the implementation of a compliant TOE, e.g., the security administrator may be refined into a Directory Administrator and a Platform Administrator.

In addition to the roles, the TOE requires the interfaces, functionality and access control to support the administration and management of the TOE. The TOE includes management capabilities to turn on or off the following security functions: security alarms, replication, and cryptomodule testing after key generation.

Through controlled access to TSF data the other TOE security functions are managed. TOEs claiming compliance to this PP may include additional management capabilities. If they also include additional functional requirements the associated management of the functions must also be considered.

**Internal Capabilities:** the TOE includes several internal security capabilities for its own protection or to support the availability of general TOE resources.  For its own protection the TOE includes requirements that relate to the integrity and management of the mechanisms that provide the TSF and to the integrity of TSF data.  These include self-testing, recovery from failure, SFP domain separation, non-bypassability of the TSP, and a reliable time-stamp.  To support the availability of required resources, the TOE requires the TSF to enforce maximum quotas on the usage of disk space, processor time, and transport-layer representation for access from a network.

# 3   TOE SECURITY ENVIRONMENT

This section discusses the characteristics of environments and threat levels appropriate for medium robustness TOEs, and it describes the specific security aspects of the environment in which the directory is intended to be used and the manner in which it is expected to be employed.  This information is provided to help organizations using this PP insure that the functional requirements specified by this medium robustness PP are appropriate for their intended application of a compliant TOE.

This section includes the following:

• Discussion of medium robustness;

• Assumptions about the security aspects of a compliant TOE environment;

• Threats to TOE assets or to the TOE environment which must be countered; and

• Organizational security policies that compliant TOEs must enforce.

## 3.1   *CHARACTERIZING MEDIUM ROBUSTNESS*

Robustness is defined as a TOE characteristic that describes how well the TOE can protect itself and its resources.  The more robust the TOE, the better it is able to protect itself.  This section relates the defining factors of the IT environment, authorization, and value of resources to the selection of appropriate robustness levels.

### 3.1.1   TOE ENVIRONMENT DEFINING FACTORS

In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: the **value of the resources** and **authorization of the entities** to access those resources.

In general terms, the environment for a TOE can be characterized by the authorizations (or lack of authorization) that the least trustworthy entity has with respect to the TOE resources with the highest value (i.e. the TOE itself and all of the data processed by the TOE).  There are an infinite number of combinations of entity authorizations and resource values since there are an infinite number of potential environments and a variety of authorizations defined by a given organization.  These two environmental factors are used in subsequent sections to assist in determining the robustness level required for in identified TOE for a given system in an environment.

### Value of Resources

The value of resources associated with a TOE is determined by the value of data being processed or used by the TOE, as well as the TOE itself in the system (for example, the directory and the role it plays supporting a PKI).  The "value" is assigned by the using organization.  For example, low-value data might be equivalent to data marked by the U.S. Government as "FOUO", while high-value data may be equivalent to data marked by the U.S. Government as "Top Secret".  In this example, a loss of life may occur if Top Secret information is compromised or if the information were unavailable

past an acceptable period of time. It is therefore considered high-valued information. In a commercial enterprise, low-value data may be an organizational structure as captured in the corporate on-line phone book, while high-value data may include corporate research results for the next generation product. In this example, millions of dollars in revenue could be lost if the research results are compromised or lost. It is therefore considered high-value information. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a directory may contain data that is available for anyone to read and has its own integrity protection (e.g., revocation lists), however if this data was updated by an unauthorized and rogue user, the authentication mechanisms that protect high value data and depend on the correctness of the revocation list could be compromised. In this example, the directory protects high value data, and therefore must be treated as a high-value part of the TOE.

## Authorization of Entities

An authorization is defined as the access control information that conveys the privileges of an entity (administrators, relying parties, other IT systems). The authorizations that entities have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) are an abstract concept that includes a combination of the trustworthiness of an entity and the access privileges granted to that entity with respect to the resources of the TOE. Some entities may hold authorizations to access all data on the TOE while others may hold minimal authorizations to access few or no TOE resources. The level of access and the abilities granted (read, modify, delete) determine the level of trust for an entity.

## 3.1.2  SELECTION OF APPROPRIATE ROBUSTNESS LEVELS

As defined above, robustness describes how well the TOE can protect itself and its resources. The more robust the TOE, the better it is able to protect itself. This section relates the defining factors of the IT environment, authorization, and value of resources to the selection of appropriate robustness levels.

When assessing any environment with regards to Information Assurance (IA), the critical point to consider is the likelihood of a compromise. This likelihood is somewhat dependent on the value of the TOE and resident data as well as logical connectivity and physical location. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase. It is critical to note that several combinations of environmental factors will result in environments in which the likelihood of an attempted compromise is similar. Consider the following two cases:

The first case is a TOE that processes low-value data. This TOE is connected to the Internet and is accessible by authorized entities. In this case, the least trusted entities are unauthorized entities exposed to the TOE as a result of Internet connectivity. Since only low-value data is being processed, the likelihood that unauthorized entities would attempt to gain access to the system is low. In this instance, TOE compliance with a basic robustness PP is sufficient.

The second case is a TOE that processes high-value information. In this example, the TOE is a stand-alone system that is both logically isolated from any external connections and is physically protected. Additionally, every entity with physical and logical access to the TOE holds the highest authorizations thereby assuring that only highly trusted users are authorized to access the TOE. In this case, even though high value information is processed, it is unlikely that a compromise of the TOE and resident information will occur simply because of the physical and logical isolation and the trustworthiness of the entities. Once again, selection of a basic robustness TOE is appropriate.

The preceding examples demonstrated that it is possible for different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. The following chart depicts the "universe" of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.

As depicted in Figure 3.1, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise", signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

While it would be possible to create many different "levels of robustness" at small intervals along the "Increasing Robustness Requirements" line to counter the increasing likelihood of attempted compromise due to those attacks, it would not be practical nor particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in Figure 3.2.
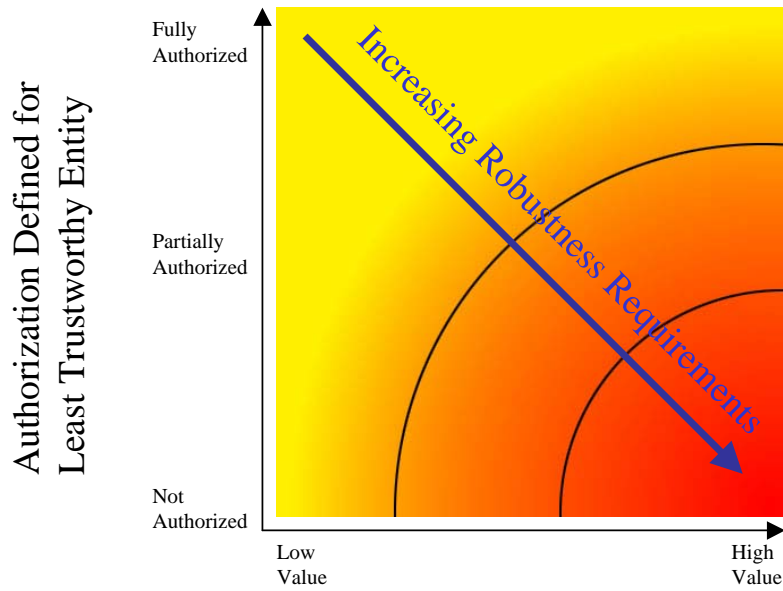
Highest Value of Resources
Associated with the TOE

**Figure 3.1 – Robustness Requirements**

In Figure 3.2 the "dots" represent given instantiations of environments; like-colored dots define environments with a similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized by like-colored dots. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a "point" in the chart above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes "low value" data vs. "medium value" data). Because every organization will be different, a rigorous definition is not possible.

Highest Value of Resources
Associated with the TOE
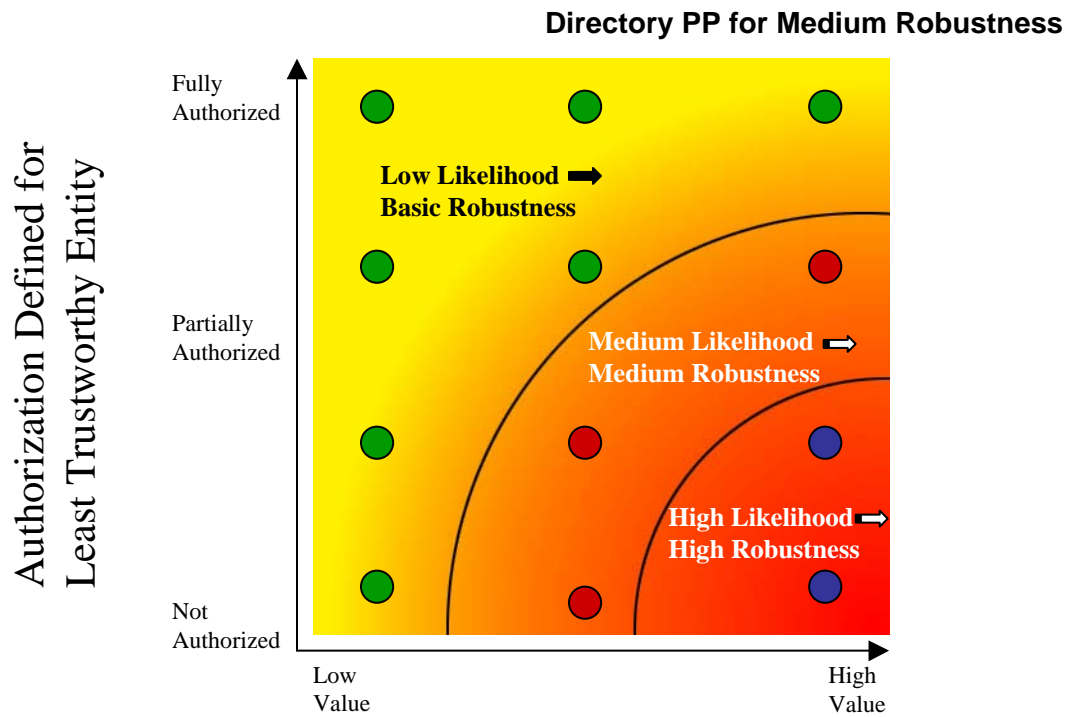
**Figure 3.2 – Robustness Levels**

### 3.1.3  Medium Robustness

Medium robustness TOEs fall in the central area of the robustness figures discussed above.  A medium robustness TOE is considered sufficient protection for environments where the likelihood of an attempted compromise is medium.  This implies that the motivation of the threat agents will be average in environments that are suitable for TOEs of medium robustness.  Note that this also implies that the resources and expertise of the threat agents really are not factors that need to be considered, because highly sophisticated threat agents will not be motivated to use great expertise or extensive resources in an environment where medium robustness is suitable.

The medium motivation of the threat agents can be reflected in a variety of ways.  One possibility is that the value of the data processed or protected by the TOE will be only medium, thus providing little motivation of even a totally unauthorized entity to attempt to compromise the data.  Another possibility, (where higher value data is processed or protected by the TOE) is that the procuring organization will provide environmental controls (that is, controls that the TOE itself does not enforce) in order to ensure that threat agents that have generally high motivation levels (because of the value of the data) cannot logically or physically access the TOE (e.g., all users are "vetted" to help ensure their trustworthiness, and connectivity to the TOE is restricted).

It is important to note to vendors and end users that any IT entity that is used to protect National Security information, and employs cryptography as a protection mechanism, will require the TOE's key management techniques to be approved by NSA when the TOE is fielded.

### *3.2 SECURE USAGE ASSUMPTIONS*

Table 3.1 lists the Secure Usage Assumptions.

**Table 3.1 – Secure Usage Assumptions**

| Assumption | Assumption Description |
|---|---|
| A.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment. |
| A.NO_GENERAL_PURPOSE | There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, web servers, database servers or user applications) available on the TOE. |
| A.REMOTE_ADUA_ENVIRONMENT | The accreditation process will ensure that the procuring organization will manage and protect the ADUA in a manner that is commensurate with this PP. |
| A.REMOTE_ADUA_FUNCTIONALITY | Remote ADUA applications are trusted applications that would comply with the security requirements of this PP that are applicable to the ADUA. |
| A.DISTIRBUTED_DIRECTORY_SECURITY_POLICY_ENFORCEMENT | Before enabling replication and/or distributed I&A mechanisms, the Security Administrator must ensure that the appropriate level of trust has been established and that the I&A and/or access control security policies are understood and enforced. |
| A.USER_INFORMATION_FLOW | Users will protect all information that is displayed or printed in accordance with both the classification of the data and local security policies. |

### *3.3 THREATS TO SECURITY*

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

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

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

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

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

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

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

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

The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE.

A threat agent's expertise and/or resources that are "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).

The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

Table 3.2 lists the threats to security.

**Table 3.2 – Threats to Security**

| Threat | Description of Threat |
|---|---|
| T. ADMIN_ ERROR | An administrator may incorrectly install or configure the TOE, or install a corrupted TOE, resulting in ineffective security mechanisms. |
| T.ADMIN_ROGUE | An administrator's intentions may become malicious resulting in user or TSF data being compromised. |
| T.AUDIT_ COMPROMISE | A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. |
| T.CORRUPTED_ IMPLEMENTATION | Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program. |
| T.CRYPTO_ COMPROMISE | A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromise the cryptographic mechanisms and the data protected by those mechanisms. |

| T.FLAWED_DESIGN | Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious  user or program. |
|---|---|
| T.MALICIOUS_TSF_ COMPROMISE | A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted). |
| T.MASQUERADE | A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources. |
| T.POOR_TEST | Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities. |
| T.REPLAY | A user may gain inappropriate access to the TOE by replaying authentication information. |
| T.RESIDUAL_DATA | A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. |
| T.RESOURCE_ EXHAUSTION | A malicious process or user may block others from system resources (e.g., CPU time) via a resource exhaustion denial of service attack. |
| T.SPOOFING | An entity may misrepresent itself as the TOE to obtain authentication data. |
| T.UNATTENDED_ SESSION | A user may gain unauthorized access to an unattended session. |
| T.UNAUTHORIZED_ ACCESS | A user may gain access to user data for which they are not authorized according to the TOE security policy. |
| T.UNIDENTIFIED_ ACTIONS | The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach. |
| T.UNKNOWN_ STATE | When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown. |

### 3.4 ORGANIZATIONAL SECURITY POLICIES

Table 3.3 lists the organizational security policies.

**Table 3.3 – Organizational Security Policies**

| Policy | Policy Description |
|---|---|
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which administrators consent by accessing the system. |
| P.ACCOUNTABILITY | The authorized users of the TOE shall be held accountable for their actions within the TOE. |
| P.ADMIN_ACCESS | Administrators shall be able to administer the TOE both locally and remotely through protected communications channels. |
| P.CRYPTOGRAPHY_ VALIDATED | Where the TOE requires FIPS-approved security functions, only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key distribution, and random number generation services). |
| P.CRYPTOGRAPHIC_ FUNCTIONS | The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. |
| P.NONREPUDIATION | The TOE must provide non-repudiation services for transmitted and received repository data.  The non-repudiation services include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that evidence of receipt the TOE is waiting for is overdue. |
| P.DISTRIBUTED_DIREC TORY_SUPPORT | Directories shall be able to support replication.  To support replication directories shall be able to replicate (both produce and consume) definable subtrees to other directories (peer trusted directories).  Directories shall be able to authenticate using a distributed authentication mechanism. |
| P.VULNERABILITY_ ANALYSIS_TEST | The TOE must undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential. |

# 4 SECURITY OBJECTIVES

This chapter describes the security objectives. These security objectives are divided between the Security Objectives for the TOE (i.e., security objectives addressed directly by the TOE), and the Security Objectives for the Operating Environment (i.e., security objectives addressed by the IT domain or by non-technical or procedural means).

## 4.1 SECURITY OBJECTIVES FOR THE TOE

Table 4.1 contains the Security Objectives for the TOE

**Table 4.1 – Security Objectives for the TOE**

| Objective | Objective Description |
|---|---|
| O.ADMIN_ROLE | The TOE will provide administrator roles to isolate administrative actions. |
| O.AUDIT_ GENERATION | The TOE will provide the capability to detect and create records of security-relevant events associated with users. |
| O.AUDIT_ PROTECTION | The TOE will provide the capability to protect audit information. |
| O.AUDIT_REVIEW | The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations. |
|  |  |
| O.CHANGE_ MANAGEMENT | The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development. |
| O.CORRECT_TSF_ OPERATION | The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. |
| O.CRYPTOGRAPHY_ VALIDATED | The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions. |
| O.CRYPTOGRAPHIC_ FUNCTIONS | The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. |
| O.DISPLAY_BANNER | The TOE will display an advisory warning regarding use of the TOE. |
| O.DOCUMENT_KEY_ LEAKAGE | The bandwidth of channels that can be used to compromise key material shall be documented. |

| O.MAINT_MODE | The TOE shall provide a mode from which recovery or initial startup procedures can be performed. |
|---|---|
| O.MANAGE | The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. |
| O.MEDIATE | The TOE must protect user data in accordance with its security policy. |
| O.NONREPUDIATION | At the option of an administrator, the TSF must be able to provide non-repudiation services for transmitted and received repository data. These services must include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that the evidence of receipt the TOE is waiting for is overdue. |
| O.REPLAY_ DETECTION | The TOE will provide a means to detect and reject the replay of authentication data. |
| O.DISTRIBUTED_DIR ECTORY_SUPPORT | The TSF shall be able to replicate definable subtrees to (produce) and accept replications of definable subtrees from (consume) other directories. The TSF shall be to authenticate using a distributed authentication mechanism. |
| O.RESIDUAL_ INFORMATION | The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. |
| O.RESOURCE_ SHARING | The TOE shall provide mechanisms that mitigate attempts to exhaust CPU time and available network connections provided by the TOE. |
| O.ROBUST_ADMIN_ GUIDANCE | The TOE will provide administrators with the necessary information for secure delivery and management. |
| O.ROBUST_TOE_ ACCESS | The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate. |
| O.SELF_ PROTECTION | The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure. |
| O.SOUND_DESIGN | The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented. |
| O.SOUND_ IMPLEMENTATION | The implementation of the TOE will be an accurate instantiation of its design, and is adequately and |

| | accurately documented. |
|---|---|
| O.THOROUGH_ FUNCTIONAL_ TESTING | The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements. |
| O.TIME_STAMPS | The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. |
| O.TRUSTED_PATH | The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. |
| O.VULNERABILITY_ ANALYSIS_TEST | The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. |

## 4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

Table 4.2 contains security objectives for the environment.

**Table 4.2 – Security Objectives for the IT Environment**

| OE.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment. |
|---|---|
| OE.NO_GENERAL_PURPOSE | There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, web servers, database servers or user applications) available on the TOE. |
| OE.REMOTE_ADUA_ENVIRONMENT | The accreditation process will ensure that the procuring organization will manage and protect the ADUA in a manner that is commensurate with this PP. |
| OE.REMOTE_ADUA_FUNCTIONALITY | Remote ADUA applications are trusted applications that would comply with the security requirements of this PP that are applicable to the ADUA. |
| OE.DISTRIBUTED_DIRECTORY_SECURITY_POLICY_ENFORCEMENT. | Before enabling replication and/or distributed I&A mechanisms, the Security Administrator must ensure that the appropriate level of trust |

| | has been established and that the I&A and/or access control security policies are understood and enforced. |
|---|---|
| OE.EVIDENCE_OF_RECEIPT_OF_REPLICA_DATA | Peer Directories must be able to provide evidence of receipt of replica data to support non-repudiation of replication activity. |
| OE.TRUSTED_PATH | Remote authorized IT entities in conjunction with the TOE must provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. |
| OE.USER_INFORMATION_FLOW | Users and Administrators will protect all information that is displayed or printed in accordance with both the classification of the data and local security policies. |

{This page intentionally left blank}

# 5  IT SECURITY REQUIREMENTS

This section provides the TOE security functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE, and the IT environment security functional requirements on which the TOE relies.  These requirements consist of functional components from Part 2 of the CC, assurance components from Part 3 of the CC, Common Criteria interpretations, NIAP interpretations, and extended functional components derived from the CC components.

## TOE Subjects and Objects

The following describes the TOE subjects and objects, and provides a basis for the security functional requirements (SFR) representation of its security services.

The subjects are the users and their internal TOE representation acting on their behalf, e.g., TOE processes.  The objects are the data in the repository of information maintained by the directory, including the entries, their attributes, and their values.

An important nuance to the definition of the objects in the TOE is that the repository data includes trusted data, i.e., TSF data.  So while the directory is responsible for controlling access to the repository data it also relies on the certificates and RLs in its repository for its own certificate-based security mechanisms, e.g., to validate signatures for authentication.

Formatting Conventions

The following formatting conventions apply to the TOE Security Functional Requirements and the Requirements for the IT Environment.

The CC allows several operations to be performed on functional requirements; *refinement, selection, assignment*, and *iteration* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement.  Refinement of security requirements is denoted by **bold text**.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement.  Selections are denoted by *italicized text*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment is indicated by showing the value in square brackets, [assignment_value].

**Application notes** provide additional information for the reader, but do not specify requirements. Application notes are denoted by *italicized text*.

The **iteration** operation is used when a component is repeated with varying operations. An iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number). (*) refers to all iterations of a component.

This PP contains several assignment and selection operations left to the ST writer to perform. The notation convention used for these is identical to that used in the Common Criteria.

## 5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

The functional security requirements for the TOE consist of the following components derived from Part 2 of the CC, CC interpretations, NIAP interpretations, and extended components, summarized in the Table 5.1 below.

**Table 5.1 – Security Functional Components**

| Functional Components | |
| --- | --- |
| FAU_ARP.1 | Security alarms |
| FAU_ARP_ACK_DIR_(EXT).1 | Extended: Security alarm acknowledgement for Directory |
| FAU_GEN.1-NIAP-0347 | Audit data generation |
| FAU_GEN.2-NIAP-0410 | User identity association |
| FAU_SAA.1-NIAP-0407 | Potential violation analysis |
| FAU_SAR.1(1) | Audit review (auditor role) |
| FAU_SAR.1(2) | Audit review (external audit analysis) |
| FAU_SAR.2 | Restricted audit review |
| FAU_SAR.3 | Selectable audit review |
| FAU_SEL.1-NIAP-0407 | Selective audit |

| Functional Components | |
|---|---|
| FAU_STG.1-NIAP-0429 | Protected audit trail storage |
| FAU_STG.3 | Action in case of possible audit data loss |
| FAU_STG.NIAP-0414-1-NIAP-0429 | Site-configurable prevention of audit data loss |
| FCO_PRA_(EXT).1(1) | Extended:  Proof of replication activity (TOE) |
| FCS_BCM_(EXT).1 | Extended:  Baseline cryptographic module |
| | |
| FCS_CKM.1(1) | Cryptographic key establishment for AES symmetric keys |
| FCS_CKM.1(2) | Cryptographic key entry for digital signature/verification private keys |
| FCS_CKM.2 | Cryptographic key distribution |
| FCS_CKM.4 | Cryptographic key destruction |
| FCS_CKM_(EXT).2 | Cryptographic Key Handling and Storage |
| FCS_COP.1(1) | Cryptographic operation (for encryption/decryption) |
| FCS_COP.1(2) | Cryptographic operation (for cryptographic signature) |
| FCS_COP.1(3) | Cryptographic operation cryptographic hashing) |
| FCS_COP.1(4) | Cryptographic operation (for cryptographic key agreement) |
| FCS_COP_(EXT).1 | Random Number Generation |
| FDD_RPL_(EXT).1 | Extended:  Replication of directory data with security attributes |
| FDP_ACC.2 | Complete access control |
| FDP_ACF.1 | Security attribute based access control (Directory Access Control SFP) |
| FDP_RIP.2 | Full residual information protection |
| FIA_AFL.1 | Authentication failure handling |
| FIA_ATD.1(1) | User attribute definition (relying party without a certificate, including anonymous access) |
| FIA_ATD.1(2) | User attribute definition (remote administrator, remote data manager, and relying party with a certificate) |
| FIA_ATD.1(3) | User attribute definition (local administrator) |
| FIA_UAU.1 | Timing of authentication (anonymous relying party) |
| FIA_UAU.2 | User authentication before any action |

| Functional Components | |
|---|---|
| FIA_UAU.5 | Multiple authentication mechanisms |
| FIA_UID.2 | User identification before any action |
| FIA_USB.1 | User-subject binding |
| FMT_MOF.1(1) | Management of security functions behaviour (directory functions) |
| FMT_MOF.1(2) | Management of security functions behaviour (cryptographic module testing) |
| FMT_MSA.1 | Management of security attributes (directory access control attributes) |
| FMT_MTD.1(1) | Management of TSF data (administration of security functions) |
| FMT_MTD.1(2) | Management of TSF data (cryptographic TSF data) |
| FMT_MTD.1(3) | Management of TSF data (time TSF data) |
| FMT_MTD.1(4) | Management of TSF data (subsets of TSF data) |
| FMT_MTD.2(1) | Management of limits on TSF data (processor time percentage) |
| FMT_MTD.2(2) | Management of limits on TSF data (transport-layer quotas) |
| FMT_SMF.1 | Specification of management functions |
| FMT_SMR.2(1) | Restrictions on security roles (strict separation) |
| FMT_SMR.2(2) | Restrictions on security roles (data administration and users) |
| FPT_ITA.1 | Inter-TSF availability within a defined availability metric |
| FPT_RCV.2 | Recovery from failure |
| FPT_RPL.1 | Replay detection |
| | |
| | |
| FPT_STM.1 | Reliable time stamps |
| FPT_TDC.1(1) | Inter-TSF basic TSF data consistency (directory time for certificate-based security mechanisms and non-repudiation services) |
| FPT_TDC.1(2) | Inter-TSF basic TSF data consistency (distinguished name character support) |
| FPT_TST_(EXT).1 | Extended:  TSF testing |

| Functional Components | |
|---|---|
| FPT_TST.1(1) | Extended:  Cryptographic testing |
| FPT_TST.1(2) | Extended:  Key Generation testing |
| FRU_RSA.1(1) | Maximum quotas (processor time) |
| FRU_RSA.1(2) | Maximum quotas (transport-layer) |
| FTA_SSL.1 | TSF-initiated session locking |
| FTA_SSL.2 | User-initiated locking |
| FTA_SSL.3(1) | TSF-initiated termination (remote administration session) |
| FTA_SSL.3(2) | TSF-initiated termination (remote directory service session) |
| FTA_TAB.1 | Default TOE access banners |
| FTA_TSE.1 | TOE session establishment |
| FTP_ITC_(EXT).1(1) | Extended:  Inter-TSF trusted channel (prevention of disclosure) |
| FTP_ITC_(EXT).1(2) | Extended:  Inter-TSF trusted channel (detection of modification) |
| FTP_TRP_(EXT).1(1) | Extended:  Trusted Path  (prevention of disclosure) |
| FTP_TRP_(EXT).1(2) | Extended:  Trusted Path (detection of modification) |

### 5.1.1  Class FAU: Security audit

For the audit functionality, the following requirements are written with the intent that the auditor is responsible for reviewing the audit trail, but the security administrator(s) is responsible for configuring the behavior of the audit mechanisms (setting thresholds, configuring which events are to be audited, etc.).

**FAU_ARP.1   Security alarms**

FAU_ARP.1.1 – **Refinement:**  The TSF shall [immediately display a message identifying the potential security violation, and make accessible the audit record contents associated with the auditable event(s) that generated the alarm, at the:
- local console;
- remote auditor and security administrator sessions that exist;
- remote auditor and security administrator sessions that are initiated before the alarm has been acknowledged; and
- [selection:  [assignment: *other methods determined by the ST author*], *"no other methods"*]]

upon detection of a potential security violation.

*Application Note:  This requirement ensures that when an event happens it's displayed to any administrator that is logged on, or queued.  The TSF provides a message to the local console regardless of whether an administrator is logged in and, to ensure administrators are aware of the alarm as soon as possible, a message is also displayed to all the remote Auditor and Security Administrator existing sessions and any new sessions until the alarm has been acknowledged.  The audit records contents associated with the alarm may or may not be part of the message displayed, however the relevant audit information must be available to both the auditor and the security administrator.*

*It is acceptable for the ST author to fill the open assignment with none, if no other methods are included in the TOE.  The following component, FAU_ARP_ACK_DIR_(EXT).1, defines the requirement for acknowledgement and notification of the acknowledgement.*

**Extended:      Security alarm acknowledgement for Directory (FAU_ARP_ACK_DIR_(EXT).1)**

FAU_ARP_ACK_DIR_(EXT).1.1 – The TSF shall display the message identifying the potential security violation and make accessible the audit record contents associated with the auditable event(s) until it has been acknowledged.

FAU_ARP_ACK_DIR_(EXT).1.2 – The TSF shall display an acknowledgement message identifying a reference to the potential security violation, a notice that it has been acknowledged, the time of the acknowledgement and the user identifier that acknowledged the alarm, at the:
- local console; and
- active sessions for remote auditor and security administrators whose sessions received the alarm.

*Application Note: This extended requirement is necessary since a CC requirement does not exist to ensure an administrator will be aware of the alarm. In FAU_ARP_ACK_DIR_(EXT).1.1,the intent is to ensure that if an administrator is logged in and not physically at the console or remote workstation the message will remain displayed until any administrator has acknowledged it. The message will not be scrolled off the screen due to other activity taking place (e.g., the Audit Administrator is running an audit report).*

*FAU_ARP_ACK_DIR_(EXT).1.2 ensures that any administrator that received the alarm message also receives notice that the alarm has been  acknowledgement.  The acknowledgment message includes some form of reference to the alarm message, who acknowledged the message and when.  FMT_MTD.1(1) requires the capability of turning on and off the alarm on an incident, and FMT_MOF.1(1) requires the capability of turning on or off the alarm function.*

**FAU_GEN.1-NIAP-0347      Audit data generation**

FAU_GEN.1.1-NIAP-0347 – **Refinement:**  The TSF shall be able to generate an audit record of the following auditable events:
- start-up and shutdown of the audit functions;
- all auditable events **listed in Table 5.2; and**
- [selection: [assignment: *events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST author*], [assignment: *events commensurate with a basic level of audit introduced by the inclusion of extended requirements determined by the ST author*], *"no additional events"*].

*Application Note:  For the selection, the ST author should choose one or both of the assignments (as detailed in the following paragraphs), or select "no additional events".  For the first assignment, the ST author augments the table (or lists explicitly) the audit events associated with the basic level of audit for any SFRs that the ST author includes that are not included in this PP.*

*Likewise, for the second assignment the ST author includes audit events that may arise due to the inclusion of any extended requirements not already in the PP.  Because "basic" audit is not defined for such requirements, the ST author will need to determine a set of events that are commensurate with the type of information that is captured at the basic level for similar requirements.  If no additional (CC or extended) SFRs are included, or if additional SFRs are included that do not have "basic" audit associated with them, then it is acceptable to assign "no additional events" in this item.*

FAU_GEN.1.2-NIAP-0347 – 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 (if applicable), and the outcome (success or failure) of the event; and

b)  for each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 5.2 below*].

*Application Note: In column 3 of the Table 5.2 below, "if applicable" is used to designate data that should be included in the audit record if it "makes sense" in the context of the event that generates the record. If no other information is required (other than that listed in "a") for a particular audit event type, then an assignment of "none" is acceptable.*

**Table 5.2 – Auditable Events**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_ARP.1 | Potential security violation was detected | Identification of what caused the generation of the alarm |
| FAU_ARP_ACK_DIR_(EXT).1 | None | The identity of the administrator that acknowledged the alarm. |
| FAU_GEN.1-NIAP-0347 | None | |
| FAU_GEN.2-NIAP-0410 | None | |
| FAU_SAA.1-NIAP-0407 | Enabling and disabling of any of the analysis mechanisms (i.e., changing the applicable rules) | The identity of the Security Administrator performing the function |
| FAU_SAR.1(1) | Opening the audit trail | The identity of the Audit Administrator performing the function |
| FAU_SAR.1(2) | Opening the audit trail | The identity of the Audit Administrator performing the function |
| FAU_SAR.2 | Unsuccessful attempts to read information from the audit records | The identity of the administrator performing the function |
| FAU_SAR.3 | None | |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_SEL.1-NIAP-0407 | All modifications to the audit configuration that occur while the audit collection functions are operating | The identity of the Security Administrator performing the function |
| FAU_STG.1-NIAP-0429 | None | |
| FAU_STG.3 | Actions taken due to exceeding the audit threshold<br><br>Fact that audit threshold was exceeded | Action taken<br><br>Percentage of storage capacity that triggered warning<br><br>The identity of the Security Administrator performing the function |
| FAU_STG.NIAP-0414-1-NIAP-0429 | None | The identity of the Security Administrator performing the function |
| FCO_PRA_(EXT).1(1) | The invocation of the non-repudiation service<br><br>When notification sent to Security Administrator that receipt acknowledgement was not received | Identity of the requestor that evidence of replication activity be generated, identification of the information, the destination, and a copy of the evidence provided. |
| FCS_BCM_(EXT).1 | None | |
| FCS_CKM.1(1) | Generation and loading of key.<br>Failure of the activity | |
| FCS_CKM.1(2) | Generation and loading of key pair for digital signatures.<br><br>Failure of the activity | |
| FCS_CKM.2 | None | |
| FCS_CKM.4 | None | |
| FCS_CKM _(EXT).1 | None | |
| FCS_CKM _(EXT).2 | None | |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_COP.1(1) | Failure of cryptographic operation | Type of cryptographic operation<br><br>Any applicable cryptographic mode(s) of operation, excluding any sensitive information |
| FCS_COP.1(2) | Failure of cryptographic operation | Type of cryptographic operation<br><br>Any applicable cryptographic mode(s) of operation, excluding any sensitive information |
| FCS_COP.1(3) | Failure of cryptographic operation | Type of cryptographic operation<br><br>Any applicable cryptographic mode(s) of operation, excluding any sensitive information |
| FCS_COP.1(4) | Failure of cryptographic operation | Type of cryptographic operation<br><br>Any applicable cryptographic mode(s) of operation, excluding any sensitive information |
| FCS_COP_(EXT).1 | Failure of cryptographic operation | Type of cryptographic operation<br><br>Any applicable cryptographic mode(s) of operation, excluding any sensitive information |
| FDD_RPL_(EXT).1 | Invocation of the replication mechanism | When TSF is the consumer: the IP address of the producer of the replica data and a reference to the unit of replication (e.g., the DN at the top of the subtree).<br><br>When TSF is the producer: the IP address of the consumer of the replica data and a reference to the unit of replication (e.g., the DN at the top of the subtree). |
| FDP_ACC.2 | None | |
| FDP_ACF.1 | All requests to perform an operation on an object covered by the SFP | The identity of the object.<br><br>The operation requested. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FDP_RIP.2 | None | |
| FIA_AFL.1 | The reaching of the threshold for the unsuccessful authentication attempts<br><br>The actions (e.g. disabling of an account) taken<br><br>The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account) | Identity of the unsuccessfully authenticated user |
| FIA_ATD.1(1) | None | |
| FIA_ATD.1(2) | None | |
| FIA_ATD.1(3) | None | |
| FIA_UAU.1 | Access to the Directory by an anonymous relying party | |
| FIA_UAU.2 | Successful and unsuccessful use of authentication mechanisms | Claimed identity of the user using the authentication mechanism, and must exclude all password information in the audit record. |
| FIA_UAU.5 | Successful and unsuccessful use of authentication mechanisms | Claimed identity of the user using the authentication mechanism, and must exclude all password information in the audit record. |
| FIA_UID.2 | All use of the user identification mechanism | Claimed identity of the user using the identification mechanism, and must exclude all password information in the audit record. |
| FIA_USB.1 | Success and failure of binding of user security attributes to a subject | The identity of the user whose attributes are attempting to be bound |
| FMT_MOF.1(*) | Enabling or Disabling a security function referenced in the associated FMT_MOF.1 components | The mechanism that was enabled/disabled<br><br>The identity of the administrator performing the function |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FMT_MSA.1 | All manipulation of the security attributes by an administrator | The old and new values of the affected security attributes<br><br>The identity of the administrator performing the function |
| FMT_MTD.1(*) | All modifications of the values of TSF data by an administrator | The old and new values of the affected TSF data<br><br>The identity of the administrator performing the function |
| FMT_MTD.2(1) | All modifications of the limits on processor time | The old and new limits<br><br>The identity of the administrator performing the function |
| FMT_MTD.2(2) | All modifications of the limits on transport-layer resources | The old and new limits<br><br>The identity of the administrator performing the function |
| FMT_SMF.1 | Use of the management functions | The identity of the administrator performing the function |
| FMT_SMR.2(*) | Modifications to the group of users that are part of a role | User IDs that are associated with the modifications, and the roles they were associated to or disassociated from<br><br>The identity of the administrator performing the function |
| FPT_ITA.1 | None | |
| FPT_RCV.2 | The fact that a failure or service discontinuity occurred<br><br>Resumption of the regular operation | Type of failure or service discontinuity |
| FPT_RPL.1 | Detect replay attack | Identity of the user that was the subject of the reply attack |
| | | |
| | | |

| Requirement | Auditable Events | Additional Audit Record Contents |
| --- | --- | --- |
| FPT_STM.1 | Changes to the time | The identity of the Administrator or Data Manager performing the function. |
| FPT_TDC.1(*) | None | |
| FPT_TST_(EXT).1 | Execution of TSF self tests and the results of the tests | The identity of the administrator performing the test, if initiated by an administrator. |
| FPT_TST.1(1) | Execution of cryptomodule self tests and the results of the tests performed | The identity of the cryptographic administrator performing the test, if initiated by an administrator |
| FPT_TST.1(2) | Execution of cryptomodule self tests and the results of the tests performed | The identity of the cryptographic administrator performing the test, if initiated by an administrator |
| FRU_RSA.1(*) | Fact that a quota was exceeded | The quota threshold that was exceeded |
| FTA_SSL.1 | Locking of an interactive session by the session locking mechanism<br><br>Any attempts at unlocking of an interactive session | The identity of the user associated with the session being locked or unlocked |
| FTA_SSL.2 | Locking of an interactive session by the session locking mechanism<br><br>Any attempts at unlocking of an interactive session | The identity of the user associated with the session being locked or unlocked |
| FTA_SSL.3(*) | The termination of a remote session by the session locking mechanism | The identity of the user associated with the session that was terminated |
| FTA_TAB.1 | None | |
| FTA_TSE.1 | All attempts at establishment of a user session | The identity of the user attempting to establish the session<br><br>For unsuccessful attempts, the reason for denial of the establishment attempt |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FTP_ITC_(EXT).1(*) | All attempted uses of the trusted channel functions | Identification of the initiator and target of the trusted channel |
| FTP_TRP_(EXT).1(*) | All attempted uses of the trusted path functions | Identification of the claimed user identity |

**FAU_GEN.2-NIAP-0410        User Identity Association**

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

*Application Note: For failed login attempts no user association is required because the user is not under TSF control until after a successful identification/authentication.*

**FAU_SAA.1-NIAP-0407        Potential violation analysis**

FAU_SAA.1.1-NIAP-0407 – The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.
FAU_SAA.1.2-NIAP-0407 – **Refinement:** The TSF shall enforce the following rules for monitoring events:
   a) accumulation of [
      1. authentication failures as defined in FIA_AFL.1(1) and FIA_AFL.1(2);
      2. [assignment: *a specified number of failed requests to access directory information within a specified time period*]];
   b) [any detected replay of authentication information or relying party operations;
   c) any detected modification of information in a trusted channel;
   d) any failure of the cryptographic self-tests;
   e) any failure of the other TSF self-tests;
   f) any detection of possible audit data loss as defined in FAU_STG.3;
   g) cryptographic administrator-specified number of encryption failures;
   h) cryptographic administrator-specified number of decryption failures;
   i) [selection: [assignment: *additional events from the set of defined auditable events*], *no additional events*]].

*Application Note: The intent of this requirement is that an alarm is generated (FAU_ARP.1) once the threshold for an event is met.  Once the alarm has been generated it is assumed that the "count" for that event is reset to zero.*

*Encryption and decryption failures occur when the cryptomodule couldn't perform the cryptographic operation, e.g., due to invalid output or memory overflow.  The failure of TSF self-tests in f) include failures of FPT_TST_(EXT). 1, FPT_TST.1(1) and FPT_TST.1(2)1.*

*Each of the lettered items above constitutes a "rule"; if the ST author wishes to specify greater functionality (for example, the triggering of multiple conditions above before an alarm is generated) the ST author should modify the assignment appropriately.*

**FAU_SAR.1(1)        Audit review (Auditor Role)**

FAU_SAR.1.1(1) – The TSF shall provide [the Auditor] with the capability to read [all audit data] from the audit records.

FAU_SAR.1.2(1) – **Refinement:** The TSF shall provide the audit records in a manner suitable for the **Auditor** to interpret the information.

*Application Note: Supporting the objective to isolate administrative actions, this requirement specifies that only the auditor is allowed to view the audit records. Please see the rationale section for more detail.*

*As specified in FAU_SAR.2, audit data is required to be available to two other security requirements. FAU_ARP_(EXT).1.1 provides the security administrator with access to audit data information related to alarms, and FAU_SAR.1(2) provides audit data to an external intrusion detection system.*

**FAU_SAR.1(2)          Audit review (External Audit Analysis)**

FAU_SAR.1.1(2) – **Refinement:** The TSF shall provide [the Data Manager for audit information] with the capability to read [all audit information] from the audit records **via [assignment:** *mechanism TSF uses to provide the audit information to the Data Manager for audit information***].**

FAU_SAR.1.2(2) – The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

*Application Note: This requirement requires that the audit data be made available to a trusted external IT entity that is granted the Data Manager Role for reading the audit information by the security administrator as specified in FMT_MTD.1(4), e.g., an external Intrusion Detection System. The ST author should fill in the assignment with the actual method used to provide the information (e.g., writing to a file, storing in the directory, available through a network service).*

**FAU_SAR.2   Restricted audit review**

FAU_SAR.2.1 – **Refinement:** The TSF shall prohibit all users read access to the audit records **in the audit trail**, except the **Auditor and the Data Manager for audit information.**

*Application Note: Audit data from the audit trail is restricted to the auditor to support isolating administrative actions, and to the data manager for audit information to support an external intrusion detection system. Also note FAU_ARP_(EXT).1.1 provides the security administrator with access to audit data information related to alarms.*

**FAU_SAR.3    Selectable audit review**

FAU_SAR.3.1 – The TSF shall provide the ability to perform *searches and sorting* of audit data, based on:
a.      [user identity;
b.      role;
c.      event type, including non-repudiation activity, replication activity;
d.      range of one or more dates;
e.      range of one or more times;
f.      objects covered by the SFP(s);
g.      success of auditable security events;
h.      failure of auditable security events;
i.      [selection: *object identity, subject identity, host identity*, *"none"*], and
j.      [selection: [assignment: *other criteria determined by the ST Author*], *"no additional criteria"*]].

*Application Note: "User identity" applies to all users; see application note for FIA_UID.2. "event type" is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.*

*It is implied that the Auditor is the only user who can perform this function since they are the only users with read access to all of the audit records in the audit trail. While the Data manager for audit information, e.g., an intrusion detection system, has access to the audit records it would not depend on the TOE to perform such operations on its behalf.*

*Audit data should be capable of being searched and sorted on all criteria specified in a – j, if applicable (i.e., not all criteria will exist in all audit records). Sorting means to arrange the audit records such that they are "grouped" together for administrative review. For example the Auditor may want all the audit records for a specified user presented together to facilitate their audit review. If no additional criteria are provided by the TOE to perform searches or sorting of audit data, the ST author selects "no additional criteria".*

**FAU_SEL.1-NIAP-0407        Selective Audit**

FAU_SEL.1.1-NIAP-0407 – **Refinement:** The TSF shall **allow only the Security Administrator** to include or exclude **at run-time** auditable events from the set of audited events based on the following attributes:
a.      *user identity;*
b.      [*role,*
c.      *event type, including non-repudiation activity, replication activity;*
d.      *objects covered by the SFP(s);*
e.      *success of auditable security events;*
f.      *failure of auditable security events,*
g.      [selection: *object identity, subject identity, host identity, "none"*], and
h.      [selection: [assignment: *list of additional criteria that audit selectivity is based upon*], *"no additional criteria"*]].

*Application Note: "User identity" applies to all users; see application note for FIA_UID.2. "event type" is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.*

**FAU_STG.1-NIAP-0429       Protected audit trail storage**

FAU_STG.1.1-NIAP-0429 – **Refinement:** The TSF shall **restrict the deletion of stored** audit records **in the audit trail to the Auditor**.

FAU_STG.1.2-NIAP-0429 – The TSF shall be able to *prevent* modifications to the audit records in the audit trail.

**FAU_STG.3   Action in case of possible audit data loss**

FAU_STG.3.1 – **Refinement:** The TSF shall [immediately alert the auditor and security administrator] if the audit trail exceeds [a security administrator-settable percentage of storage capacity].

**FAU_STG.NIAP-0414-1-NIAP-0429 Site-configurable Prevention of audit data loss**

FAU_STG.NIAP-0414-1.1-NIAP-0429 – **Refinement:** The TSF shall provide **a Security** Administrator **with** the capability to select one or more of the following actions *prevent auditable events, except those taken by the authorized user with special rights, overwrite the oldest stored audit records* and [selection: [assignment: *other actions to be taken in case of audit storage failure]*, *"no additional options"*] to be taken if the audit trail is full.

FAU_STG.NIAP-0414-1.2-NIAP-0429 – The TSF shall [selection: <u>choose one of:</u> *'ignore auditable events', 'prevent auditable events, except those taken by the authorized user with special rights', 'overwrite the oldest stored audit records'*] and [assignment: *other actions to be taken in case of audit storage failure]*] if the audit trail is full and no other action has been selected.

*Application Note: The TOE provides the Security Administrator the option of preventing audit data loss by preventing auditable events from occurring. The Security Administrator's actions under these circumstances are not required to be audited. The TOE also provides the Security Administrator the option of overwriting "old" audit records rather than preventing auditable events, which may protect against a denial-of-service attack. Note that this last capability technically conflicts with FAU_STG.1-NIAP-0429, which specifies that the TOE should restrict deletion to the Auditor. From the perspective of mitigating the threat that the audit trail is compromised, however, these two requirements do not conflict and can co-exist; see the rationale section for more detail.*

*As specified in the Annex information for this requirement, the second element provides a default, but if the administrator has chosen an option then that's the one that must be enforced.*

*The ST author should fill in other technology-specific actions that can be taken for audit storage failure (in addition to the two already specified), or select "no additional options" if there are no such technology-specific actions.*

*Application Note: The naming conventions are inconsistent between NIAP Interpretation I-0414 and I-0429, This component uses the labeling specified in NIAP Interpretation I-0429.*

### 5.1.2  Class FCO:  Communication

The following extended requirement for non-repudiation of replication activity includes functions to support the Directories' role in a PKI to provide non-repudiation for the replication process required by FDD_RPL_(EXT).1.  The non-repudiation service verifies the replication process was successful by generating evidence that the replica data was sent and received without error, as well as provides proof of the originator and recipient of the replicated data.  The requirement includes both the generation and verification of evidence for non-repudiation, including timestamps for when the replica data was sent and when its been received (as reported by the consumer), and notification when evidence of receipt the TOE is waiting for is overdue.

**Extended: Proof of Replication Activity (FCO_PRA_(EXT).1(1))**

FCO_PRA_(EXT).1.1(1) – The TSF shall be able to generate evidence of origin for transmitted Replica Data that consists of the identity of the IT Entity originating the activity, the fact that replication activity was initiated, the time that the activity was initiated, and [assignment: *other information included bound as "evidence of origin"*].

*Application Note: This applies when the TOE is the originator of the replication activity (defined in FDD_RPL_(EXT).1). The intent is that evidence be produced that will prove that the TOE originated a replication event at a certain time. The assignment should be used by the ST author to specify any other information that will be included (and presumably signed by the TOE) as evidence of the initiation of a replication event activity (for instance, the name of the sub-hierarchy to be replicated).*

FCO_PRA_(EXT).1.2(1) – The TSF shall be able to generate evidence of receipt for Replica Data received from other IT entities that consists of the identity of the IT Entity receiving the activity, the fact that the replication data were received, the time of receipt, and [assignment: *other information included bound as "evidence of receipt"*].

*Application Note: This applies when the TOE is the receiver of the replication activity (defined in FDD_RPL_(EXT).1). The intent is that evidence be produced that will prove that the TOE received replication data at a certain time. The assignment should be used by the ST author to specify any other information that will be included (and presumably signed by the TOE) as evidence of the receipt of a replication event activity (for instance, the name of the sub-hierarchy to be replicated).*

FCO_PRA_(EXT).1.3(1) – The TSF shall produce and maintain "evidence of replication activity" that binds, in a way that cannot be repudiated, the evidence of origin and the evidence of receipt to all fields of the replica data.

*Application Note: For non-repudiation of replication data the requirement to relate the identity to all the fields can be satisfied using replication agreement configuration information and similar bulk loading specifications. The intent of this requirement is to provide non-repudiation that the replication process was received, processed, and completed without error, and that the evidence used is not ephemeral. It should be noted that in order to meet this requirement the source of the replica data will have to have some means to accept the evidence of receipt from the consumer.*

FCO_PRA_(EXT).1.4(1) – When originating a replication activity, the TSF shall be able to send notification using [assignment: *mechanism(s)*] to a Security Administrator if it does not receive evidence of receipt of transmitted Replica Data within a Security Administrator-specified time period.

*Application Note: The assignment should be filled in with the mechanism or mechanisms used to send the notification to the Security Administrator; this could be e-mail, a message to the console, etc.*

FCO_PRA_(EXT).1.5(1) – The TSF shall provide a capability for a Security Administrator and [selection: [assignment: *other roles*], *"no other roles"*] to verify the evidence of replication activity.

*Application Note: The assignment should be filled in with the roles that are authorized to perform the actions required to verify information about a replication event. The "evidence of replication activity" is specified in FCO_PRA_(EXT).1.3.*

### 5.1.3   Cryptographic Support (FCS)
*This section specifies the cryptographic support required in the TOE. Evolving public standards on cryptographic functions and related areas have required an interim approach*

*to writing cryptographic requirements. These cryptographic requirements are expected to be achievable in commercial products in the near term, and gradually mature over time. Today these requirements represent a step in the direction of helping to improve the security in COTS products. Over time, the Protection Profile will be updated as the underlying public standards and the body of related special publications mature.*

Extended: Baseline Cryptographic Module (FCS_BCM_(EXT))
*The cryptographic requirements are structured to accommodate use of the FIPS 140-2 standard and NIST's Cryptomodule Validation Program (CMVP) in meeting the requirements. Note that FIPS-approved cryptographic functions are required to be implemented in a FIPS-validated module running in FIPS-approved mode. FCS_BCM reflects this requirement, and it specifies the required FIPS validation levels for the security functions. Note also that some of the requirements of this Protection Profile go beyond what is required for FIPS 140-2 validation.*

> *Application Note: A FIPS-approved cryptographic function is a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.*

Extended: Baseline Cryptographic Module (FCS_BCM_(EXT).1)

FCS_BCM_(EXT).1.1 **All FIPS-approved cryptographic functions implemented by the TOE shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions implemented by the TOE.**

*Application Note: This Protection Profile shall use the term "FIPS 140-2" for simplicity. FIPS PUB 140-2 is currently undergoing a regular five year review; in the near future, FIPS PUB 140-3 will supersede it. Security Targets written to comply with this Protection Profile may replace it with the successor standard that is in force at the time of evaluation.*

*Application Note: This requirement does not preclude additional cryptographic algorithms from being implemented in the cryptomodule, and/or used by the TOE for purposes OTHER than those explicitly stated in this Protection Profile.*

FCS_BCM_(EXT).1.2 **All cryptographic modules implemented in the TOE** *[selection:*

> (1) Entirely in hardware shall have a minimum overall rating of FIPS PUB 140-2, Level 3,
>
> (2) Entirely in software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS PUB 140-2, Level 3 for the following: Cryptographic Module Ports and Interfaces; Roles, Services and Authentication; Cryptographic Key Management; and Design Assurance.
>
> (3) As a combination of hardware and software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS PUB 140-2, Level 3 for the following: Cryptographic Module Ports and Interfaces; Roles, Services and Authentication; Cryptographic Key Management; and Design Assurance. *]*

*Application Note: "Combination of hardware and software" means that some part of the cryptographic functionality will be implemented as a software component of the TSF. The combination of a cryptographic hardware module and a software device driver whose sole purpose is to communicate with the hardware module is considered a hardware module rather than "combination of hardware and software".*

*Application Note: Note that the requirements for selections (2) and (3) are the same. The ST author should make it clear how the cryptomodule is implemented.*

## Cryptographic Key Management (FCS_CKM)
*NIST Special Publication 800-57, "Recommendation for Key Management" contains additional protection mechanisms that vendors are encouraged to implement. It should also be used as guidance for the cryptographic key management requirements.*

Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(1))

**FCS_CKM.1.1(1)** Refinement: **The TSF shall generate symmetric cryptographic keys** using a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and provide integrity protection to generated symmetric keys in accordance with NIST SP 800-57 "Recommendation for Key Management" Section 6.1.

*Application Note: NIST SP 800-57 "Recommendation for Key Management" Section 6.1 states: "Integrity protection can be provided by cryptographic integrity mechanisms (e.g. cryptographic checksums, cryptographic hashes, MACs, and signatures), non-cryptographic integrity mechanisms (e.g. CRCs, parity, etc.) […], or physical protection mechanisms." Guidance for the selection of appropriate integrity mechanisms is given in Sections 6.2.1.2 and 6.2.2.2 of NIST SP 800-57 "Recommendation for Key Management".*

*Application Note: Note that there is a separate requirement for Cryptographic Key Agreement (FCS_COP.1(4)).*

Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(2))

FCS_CKM.1.1(2) Refinement: **The TSF shall generate** asymmetric **cryptographic keys in accordance** with the mathematical specifications of the FIPS-approved or NIST-recommended standard [*assignment: specify standard(s)],* using a domain parameter generator and *[selection:*

> (1) a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and/or

> (2) a prime number generator as specified in ANSI X9.80 "Prime Number Generation, Primality Testing, and Primality Certificates" using random integers with deterministic tests, or constructive generation methods ]

> *in a cryptographic key generation scheme that meets the following:*

> ▪ **The TSF shall provide integrity protection and assurance of domain parameter and public key validity to generated asymmetric keys in accordance with NIST SP 800-57 "Recommendation for Key Management" Section 6.1.**
> ▪ **Generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates.**

*Application Note: NIST SP 800-57 "Recommendation for Key Management" Section 6.1 states: "Integrity protection can be provided by cryptographic integrity mechanisms (e.g. cryptographic checksums, cryptographic hashes, MACs, and signatures), non-cryptographic integrity mechanisms (e.g. CRCs, parity, etc.) […], or physical protection mechanisms." Guidance for the selection of appropriate integrity mechanisms is given in Sections 6.2.1.2 and 6.2.2.2 of NIST SP 800-57 "Recommendation for Key Management".*

*Application Note: Assurance of domain parameter and public key validity provides confidence that the parameters and keys are arithmetically correct. Guidance for the selection of appropriate validation mechanisms is given in NIST SP 800-57 "Recommendation for Key Management," NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography," and FIPS PUB 186-3, "Digital Signature Standard."*

*Application Note: See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.*

Cryptographic Key Distribution (FCS_CKM.2)

**FCS_CKM.2.1** **The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method** *[selection:*

> (3) Manual (Physical) Method, and/or

> (4) Automated (Electronic) Method ]

**that meets the following:**

> ▪ **NIST Special Publication 800-57, "Recommendation for Key Management" Section 8.1.5**
> ▪ **NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**

*Application Note: NIST Special Publication 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" is only applicable when public key schemes are used in key transport methods.*

*Application Note: DoD applications may have additional key distribution requirements related to the DoD PKI and certificate formats.*

Extended: Cryptographic Key Handling and Storage (FCS_CKM_(EXT).2)

**FCS_CKM_(EXT).2.1** **The TSF shall perform a key error detection check on each transfer of key (internal, intermediate transfers).**

*Application Note: A parity check is an example of a key error detection check.*

**FCS_CKM_(EXT).2.2** **The TSF shall store <u>persistent</u> <u>secret and private</u> keys <u>when not in use</u> in encrypted form or using split knowledge procedures.**

*Application Note: Note that this requirement is stronger than the FIPS 140-2 key storage requirements, which state: "Cryptographic keys stored within a cryptographic module shall be stored in plaintext form or encrypted form."*

*Application Note: A persistent key, such as a file encryption key, is one that must be available in the system over long periods of time. A non-persistent key, such as a key used to encrypt or decrypt a single message or a session, is one that is ephemeral in the system.*

*Application Note: "When not in use" is interpreted in the strictest sense so that persistent keys only exist in plaintext form during intervals of operational necessity. For example, a file encryption key exists in plaintext form only during actual encryption and/or decryption processing of a file. Once the file is decrypted or encrypted, the file encryption key should immediately be covered for protection.*

*Application Note: A "split knowledge procedure" is a process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, which can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.*

**FCS_CKM_(EXT)_2.3** **The TSF shall destroy non-persistent cryptographic keys after a cryptographic administrator-defined period of time of inactivity.**

*Application Note: The cryptographic administrator must have the ability to set a threshold of inactivity after which non-persistent keys must be destroyed in accordance with FCS_CKM.4.*

**FCS_CKM_(EXT).2.4** **The TSF shall prevent archiving of expired (private) signature keys.**

*Application Note: This requirement is orthogonal to typical system back-up procedures. Therefore, it does not address the problem of archiving an active (private) signature key during a system back-up and saving the key beyond its intended life span.*

Cryptographic Key Destruction (FCS_CKM.4)

*Application Note: Note that this requirement is stronger than the FIPS 140-2 key zeroization requirements, which state: "A cryptographic module shall provide methods to zeroize all plaintext secret and private cryptographic keys and CSPs within the module."*

**FCS_CKM.4.1** Refinement: **The TSF shall destroy cryptographic keys in accordance with a** cryptographic key zeroization method **that meets the following:**

> a) **Key zeroization requirements of FIPS PUB 140-2, "Security Requirements for Cryptographic Modules"**
>
> b) **Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete.**

*Application Note: The term "immediate" here is meant to impart some urgency to the destruction: it should happen as soon as practical after the key is no longer required to be in plaintext. It is certainly permissible to complete a critical section of code before destroying the key. However, the destruction shouldn't wait for idle time, and there shouldn't be any non-determined event (such as waiting for user input) which occurs before it is destroyed.*

> c) **The TSF shall zeroize each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/critical cryptographic security parameter to another location.**

*Application Note: Item c) pertains to the elimination of internal, temporary copies of keys/parameters during processing, and not to the locations that are used for the storage of the keys, which are specified in item b). The temporary locations could include memory registers, physical memory locations, and even page files and memory dumps.*

> d) **For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time.**

*Application Note: Although verification of the zeroization of each intermediate location consisting of non-volatile memories is desired here (by checking for the final known alternating data pattern), it is not required at this time. However, vendors are highly encouraged to incorporate this verification whenever possible into their implementations.*

> e) **For volatile memory and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify.**

## Cryptographic Operation (FCS_COP)

Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

**FCS_COP.1.1(1)** Refinement: **The cryptomodule shall perform** encryption and decryption using the FIPS-approved security function AES algorithm operating in *[assignment: one or more FIPS-approved modes]* and cryptographic key size of [*selection: one or more of 128 bits, 192 bits, 256 bits*]**.**

Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

**FCS_COP.1.1(2)** Refinement: **The TSF shall perform** cryptographic signature services using the FIPS-approved security function *[selection:*

> (5) Digital Signature Algorithm (DSA) with a key size (modulus) of [assignment: 2048 bits or greater],
>
> (6) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [assignment: 2048 bits or greater], or

(7) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of [selection: one or more of 256 bits, 384 bits, 521 bits], using only the NIST curve(s) [selection: one or more of P-256, P-384, P-521 as defined in FIPS PUB 186-3, "Digital Signature Standard"] ]

that meets NIST Special Publication 800-57, "Recommendation for Key Management."

*Application Note: For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for key exchange, elliptic curves will be required after all the necessary standards and other supporting information are fully established.*

Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

**FCS_COP.1.1(3)** Refinement: **The TSF shall perform** cryptographic hashing services using the FIPS-approved security function Secure Hash Algorithm and message digest size of *[selection: one or more of 256 bits, 384 bits, 512 bits].*

*Application Note: The message digest size should correspond to double the system symmetric encryption key strength.*

Cryptographic Operation (for cryptographic key agreement) (FCS_COP.1(4))

*Application Note: "Cryptographic key agreement" is a procedure where the resultant secret keying material is a function of information contributed by two participants, so that no party can predetermine the value of the secret keying material independently from the contributions of the other parties.*

**FCS_COP.1.1(4)** Refinement: **The TSF shall perform** cryptographic key agreement services using the FIPS-approved security function  as specified in NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" *[selection:*

> **(1) [assignment: Finite Field-based key agreement algorithm] and cryptographic key sizes (modulus) of [assignment: 2048 bits or greater], or**
> **(2) [assignment: Elliptic Curve-based key agreement algorithm] and cryptographic key size of [assignment: one or more of 256 bits, 384 bits, 521 bits], using only the NIST curve(s) [selection: one or more of P-256, P-384, P-521 as defined in FIPS PUB 186-3, "Digital Signature Standard"] ]**

*Application Note: For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for key exchange, elliptic curves will be required after all the necessary standards and other supporting information are fully established.*

> **that meets NIST Special Publication 800-57, "Recommendation for Key Management."**

*Application Note: Some authentication mechanism on the keying material is recommended. In addition, repeated generation of the same shared secrets should be avoided.*

*Application Note: FIPS 140-2 Annex D specifies references for FIPS-approved Key Establishment Techniques, one of which is NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography."*

Extended: Random Number Generation (FCS_COP_(EXT).1)

FCS_COP_(EXT).1.1 The TSF shall perform all random number generation (RNG) services in accordance with a FIPS-approved RNG [*assignment: one of the RNGS specified in FIPS 140-2 Annex C*] seeded by *[selection:*

> **(1) one or more independent hardware-based entropy sources,**

*and/or*

*(2) one or more independent software-based entropy sources, and/or*

*(3) a combination of hardware-based and software-based entropy sources. ]*

*Application Note: The ST author should specify how the RNG is seeded.*

FCS_COP_(EXT).1.2 **The TSF shall defend against tampering of the random number generation (RNG)/ pseudorandom number generation (PRNG) sources.**

*Application Note: The RNG/PRNG should be resistant to manipulation or analysis of its sources, or any attempts to predictably influence its states. Three examples of very different approaches the TSF might pursue to address this include: a) identifying the fact that physical security must be applied to the product, b) applying checksums over the sources, or c) designing and implementing the TSF RNG with a concept similar to a keyed hash (e.g., where periodically, the initial state of the hash is changed unpredictably and each change is protected as when provided on a tamper-protected token, or in a secure area of memory.*

### 5.1.4   Class FDD: Directory Functions

**Extended: Replication of directory data with security attributes (FDD_RPL_(EXT).1)**

*Application Note:  This component requires a replication function to increase the availability of the Directory's repository data within a system.  A Directory provides access to information, including authentication information such as certificates and RLs.  By replicating the repository information to other directories and receiving replicated repository information in a manner that ensures the integrity of the data and its associated security attributes, the availability of the data for the system is increased.*

*The terms supplier and consumer are used to identify the source and destination of replication updates, respectively. A supplier Directory sends updates to a consumer directory, and a consumer directory accepts those updates.  The TOE includes both consumer and producer functions.  The repository data that is replicated is referred to as the 'replica'.  The configuration information for replication is referred to as the replication agreement, specified in FMT_MTD.1(1).*

*This extended component is necessary to specify a unique requirement for a directory service that is not addressed by the CC.  The requirement incorporates elements of FDP_ITC.2, FDP_ETC.2, FPT_TDC.1, and elements unique to this security service.*

FDD_RPL_(EXT).1.1 – The TSF shall support a replication mechanism for exporting and importing security administrator-defined replica data to Security Administrator-specified Peer Trusted Directories assigned as Data Managers.

FDD_RPL_(EXT).1.2 – The TSF shall export and import the data with all associated security attributes.

*Application Note:  All the security attributes must be included with all replicated data.   For example this includes all 'inherited' ACIs, e.g.,  Directory Access Control Domain ACI entries.*

FDD_RPL_(EXT).1.3 – The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported data.

FDD_RPL_(EXT).1.4 – The TSF shall ensure that the security attributes, when imported from outside the TSC, are unambiguously associated with the imported data.

FDD_RPL_(EXT).1.5 – The TSF shall provide the capability to consistently interpret the security attributes associated with the data.

*Application Note: The ST author when describing the replication function should specify the following in the TOE Summary Specification:*

• the interfaces available in sufficient detail to identify protocols, standard or proprietary, and

• identify the mechanisms used to enforce consistency, e.g., compliance with a replication standard.

## 5.1.5  Class FDP: User Data Protection

The access control decisions are based on the access rights defined for the repository information managed by the directory.  These rights are defined in an access control specification, referred to as an ACI.  The ACI grants or denies permission to repository information in regard to a set of specified users and protected items.  The scope of the protected items can be a specific attribute, a single entry or a collection of entries.  A single ACI can apply to multiple data items, or even entire subtrees within the directory repository.  Multiple ACIs can apply to any given data item.

To enforce access controls, the directory first identifies all access control specifications that apply to the target of the directory operation.  Then, it determines whether any of these specifications apply to the requestor of the operation.  If so, it enforces any access control specifications that apply to that particular requestor.

When the repository information is replicated between directory servers, all applicable access control information must also be replicated so that the consumer directory server of the replica can consistently enforce security and access control policies.  ST authors are required to specify in FDD_RPL_(EXT).1 how the product ensures that all applicable access control policies can be propagated.

### FDP_ACC.2   Complete access control

*Application Note: While multiple access control policies are allowed, compliant TOEs are only required to implement one access control policy.*

*For TOEs with multiple access control policies, an ST author should iterate FDP_ACF.1, and if applicable, FDP_ACC.  In addition, if an ST author wants to include support for multiple policies operating concurrently on the repository information, the ST author must identify in FDP_ACF.1.2 how the TOE knows which policy to apply.*

FDP_ACC.2.1 – The TSF shall enforce the [Directory Access Control SFP] on [
  - Subjects: Data Manager, Relying Party;
  - Objects: repository information entry, repository information attribute type, repository information attribute value, [selection: [assignment: *other directory objects*], *"none"*];]

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 – The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

*Application Note: In the first selection, the ST author should identify other objects on which access control is applied, and make appropriate changes to FDP_ACF.1(1) to reflect this addition. If no other objects are supported the ST author should select "none".*

## FDP_ACF.1   Security attribute based access control (Directory Access Control SFP)

FDP_ACF.1.1 – The TSF shall enforce the [Directory Access Control SFP] to objects based on the following: [
  a) Subject security attributes:
      - Distinguished Name,
      - User Group,
      - Role,
      - Authentication level,
      - [selection: [assignment: *other* ], *"none"*];

*Application Note: Authentication level refers to how the subject authenticated to the directory: anonymously, with a password, or with a certificate.*

*It's CC convention that the requested operation is an implicit subject attribute.*

*Access control decisions based on a subject's domain may be implemented with the User Group attribute.*

*In an implementation the role may be defined by the method by which the user accesses the TOE, as opposed to an extended "attribute" maintained by the TSF by the user.*

*Access control decisions are based on a users identity via distinguished name attribute. Through this requirement an implementation may prevent access from anonymous users and an ST author may want to describe any features that facilitates this in an access control policy.*

  b) Object security attributes:
      - Access control information (ACI) item(s) each specifying the following:
          o objects for which the ACI applies,

*Application Note: TOE's that implement 'hierarchical control' e.g., Directory Access Control Domains (DACDs), should represent this functionality as an additional refinement for specifying the ACIs that apply to an object.*

    o   subjects for which the ACI applies,
    o   priority of the ACI,
    o   access allowed or denied,
    o   authentication level required,
    o   [selection: [assignment: *other* ], *"none"*]].

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

    a)  the set of all 'associated ACIs' must be considered.
- the set of all 'associated ACIs' must include both ACIs assigned to the requested object.
- the set of all 'associated ACIs' must include ACIs where:
    - o  the subject requestor (distinguished name, user group, role) is authenticated at the required level and is in the ACI subject's set;
    - o  the protected object of the operation is in the ACI objects set;
    - o  [selection: [assignment: *other, e.g., scope of influence when the TOE*
    - o  *supports multiple concurrent access control policies*], *"none"*];
- the set of all 'associated ACIs' are established using the following algorithm [assignment: *algorithm*].

    b)  the access control decision must apply the following rules to the 'associated ACIs':
- only ACIs with the highest priority are considered;
- if priority is equal then only the ACIs with the most specific subjects are considered;
- if priority and most specific subject are equal then only the ACIs with the most specific objects are considered;
- grant access only if all access control decision ACIs grant access, i.e., if there are no ACIs, or at least one of them denies access, then access is denied.
- [selection: [assignment: *other*], *"none"*];

    c)  the access control decision is made using the following algorithm [assignment: *algorithm*]].

FDP_ACF.1.3 – The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

    a)  [assignment: *additional rules, based on security attributes that explicitly grant access of subjects to objects*].

FDP_ACF.1.4 – The TSF shall explicitly deny access of subjects to objects based on the [rules:

    a)  Relying parties are denied all access except read access;

*Application Note: The ST author should explicitly state in the TSS how this requirement would be met, e.g., using standard ACIs, it's hard coded, etc.*

    b)  [assignment: *additional rules, based on security attributes, that explicitly deny access of subjects to objects*]].

*Application Note:  This requirement applies CCIMB 0103.*

**FDP_RIP.2    Full residual information protection**

FDP_RIP.2.1 – The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects.

### 5.1.6   Class FIA: Identification and Authentication

TOE security functions implemented by a probabilistic or permutational mechanism (e.g., password or hash function) are required (at EAL2 and higher).

**FIA_AFL.1    Authentication failure handling**

FIA_AFL.1.1– **Refinement:** The TSF shall detect when *a Security Administrator  configurable positive integer within* [assignment*:  range of acceptable values*] unsuccessful authentication attempts occur related to [Security Administrator attempts to authenticate remotely, and all Auditor, Crypto Administrator, Data Manager, and relying party authentication attempts].

*Application Note: This requirement does not apply to Security administrator local authentication attempts, since it does not make sense to lock a local security administrator's account in this fashion. This could be addressed by requiring a separate account for local security administrators, which would be stated in the administrative guidance, or the TOE's authentication mechanism implementation could distinguish login attempts that are made locally and remotely.*

FIA_AFL.1.2 – When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [prevent the remote security administrator, Auditors, Crypto Administrators, Data Managers, and relying parties from performing activities that require authentication until an action is taken by the Security Administrator].

*Application Note:  This requirement applies CCIMB 0111.*

*If a product has multiple mechanisms controlled by different administrators e.g., authentication to the platform vs. authentication to the Directory, then the ST author should iterate this component as appropriate for their product.*

**FIA_ATD.1(1)User attribute definition (Relying Party without a certificate, including anonymous access)**

FIA_ATD.1.1(1) – **Refinement:** The TSF shall maintain the following list of security attributes belonging to **relying parties without certificates and anonymous relying parties** : [
   a) user identifier;
   b) role;
   c) type of authentication;
   d) user group;
   e) [selection: [assignment: *other attributes for a user as defined by the ST author*], *"none"*]].

*Application Note: The ST author should be more specific with respect to the user identifier if possible.  For example, GDS requires that the EDI_PI be employed to uniquely identify individuals, organizations, devices, and locations, so the ST author might choose to use "EDI_PI" instead of "user identifier" in "a" above.*

*In an implementation the role may be defined by the method by which the user accesses the TOE, as opposed to an explicit "attribute" maintained by the TSF by the user.*

*For anonymous relying parties, the TOE will "fill in" the attributes based on those applicable for anonymous access to the TOE.*

*"Type of authentication" is used to indicate which authentication method is to be used for the user if the TOE supports multiple authentication mechanisms, and also may be used in access control decisions (e.g., a user logging on with a password may access a subset of the objects accessible to users logging on using a certificate).*


**FIA_ATD.1(2)User attribute definition (Remote Administrator, Remote Data Manager, and Relying Party with a certificate)**

FIA_ATD.1.1(2) – **Refinement:** The TSF shall maintain the following list of security attributes belonging to **remote administrators, remote data managers, and relying parties with certificates**: [
   a) user identifier;
   b) role;
   c) type of authentication;
   d) X.509 public key certificate;
   e) user group;
   f) [selection: [assignment: *attributes associated with certificates*], *"no other attributes specified by certificates"*];
   g) [selection: [assignment: *other attributes for a user as defined by the ST author*] *"none"*]].

*Application Note: In addition to humans, this type of user could also be a trusted IT entity that performs some administrative function on the directory.*

*The ST author should be more specific with respect to the user identifier if possible.  For example, GDS requires that the EDI_PI be employed to uniquely identify individuals, organizations, devices, and locations, so the ST author might choose to use "EDI_PI" instead of "user identifier" in "a" above.  Similarly for a trusted IT entity the identifier may be the IP address and port.*

*In an implementation the role may be defined by the method by which the user accesses the TOE, as opposed to an explicit "attribute" maintained by the TSF by the user.*

*"Type of authentication" is used to indicate which authentication method is to be used for the user if the TOE supports multiple authentication mechanisms, and also may be used in access control decisions (e.g., a user logging on with a password may access a subset of the objects accessible to users logging on using a certificate).*

*In "f", the ST author should identify certificates (e.g., "attribute certificates") that are used by the TOE in making security decisions. If no certificates other than X.509 public key certificates are used, the ST author should select "none".*

**FIA_ATD.1(3)User attribute definition (Local Administrator)**

FIA_ATD.1.1(3) – **Refinement:** The TSF shall maintain the following list of security attributes belonging to **local administrators**: [
   a) user identifier(s);
   b) role;
   c) [selection: [assignment: *other attributes for a user as defined by the ST author*], *"none"*]].

*Application Note: In addition to humans, this type of user could also be a trusted IT entity that performs some administrative function on the platform.*

*In an implementation the role may be defined by the method by which the user accesses the TOE, as opposed to an explicit "attribute" maintained by the TSF for the user. Therefore, "local administrators" refer to all roles when they are invoked locally (on the machine, as opposed to remote invocation over the network).*

*This iteration of the FIA_ATD component should be used by ST authors to capture the attributes for parts of the TOE (other than the directory application) that require administrative access (for example, the Operating System on which the directory application runs). While certificate-based authentication for platform administrators is not required by this PP, if a platform implements a certificate-based mechanism the ST author should specify this attributes similar to those in FIA_ATD.1(2) in element "c" of this component.*

**FIA_UAU.1    Timing of authentication (anonymous Relying Party)**

FIA_UAU.1.1 – **Refinement:** The TSF shall allow [access to directory information base objects in accordance with the defined access control policy for anonymous users] on behalf of **anonymous relying parties** 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.

**FIA_UAU.2    User authentication before any action**

FIA_UAU.2.1 – **Refinement:** The TSF shall require each **Administrator, Data Manager, and authenticated Relying Party** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

### FIA_UAU.5    Multiple authentication mechanisms

FIA_UAU.5.1 – **Refinement:** The TSF shall provide [
   a) password,
   b) one-way certificate-based,
   c) two-way certificate-based,
   d) [assignment:  *distributed authentication mechanism*],
   e) [assignment: *other authentication mechanisms*]]
**mechanisms** to support user authentication.


*Application Note:  The intent of this requirement PP is that a compliant TOE must provide the authentication mechanism, i.e., the entire mechanism must be within the TOE.  While this is a widely accepted interpretation of the requirement there is still debate in the community, this application note clarifies the PP's intent for this requirement.*

*The distributed authentication mechanism is required to support distributed directory operations.  The PP does not specify the mechanism and relies on requirement's role in meeting the various TOE objectives that depend on the authentication mechanism to ensure the assignment is met with an adequate mechanism.  Two examples of distributed authentication mechanisms that a compliant TOE may implement are '3rd party introduction' and '3rd party presentation'.  '3rd Party introduction' trusts that the peer directory correctly verified the authentication credentials of the relying party before passing the chained request to the TOE.  '3rd Party presentation' trusts that the peer directory ensured the integrity and, if necessary, the confidentiality of the authentication credentials passed to the TOE as part of the chained request.  Both these mechanisms require that trust is established with the peer directory.*


FIA_UAU.5.2 – The TSF shall authenticate any user's claimed identity according to the [following rules:
   a) [selection: [assignment*: local administrator(s) and local data managers as defined by the ST author*], *"none"*] shall use the password mechanism;
   b) non-anonymous Relying Party authenticating without a certificate shall use the password mechanism;
   c) remote Administrator and Remote Data Manager shall use security-administrator-specified one-way or two-way certificate-based authentication, performed as described in FCS_COP_(EXT).3;
   d) relying Party with a certificate shall use the one-way certificate-based authentication as described in FCS_COP_(EXT).3;
   e) a Relying Party may be considered authenticated by a distributed authentication mechanism via a Data Manager that used certificate-based authentication (item c above) and is trusted to participate in the distributed authentication process;
   f) [selection: [assignment: *other rules as defined by the ST author*], *"none"*]].


*Application Note:  For the first selection in element FIA_UAU.5.2, the ST author should fill in the assignment for the administrators and data managers, (e.g., cryptographic administrator, security administrator) which use passwords to authenticate when they access the TOE locally.  If they use another mechanism (e.g., certificates), then "none" should be selected and if necessary, the appropriate assignment be made in item "f".*

*The ST author when describing distributed authentication in the TOE Summary Specification should specify the available interfaces in sufficient detail to facilitate system architecture and interoperability issues, e.g., identify the protocols and whether they are standard or proprietary.*

**FIA_UID.2     User identification before any action**

FIA_UID.2.1 – The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

*Application Note: This component applies to all users (administrators, and relying parties). Because of the nature of connections to the directory, even anonymous relying parties are identified (as "anonymous") prior to performing any actions on the TOE.*

**FIA_USB.1     User-Subject Binding**

FIA_USB.1.1– **Refinement:** The TSF shall associate **all** user security attributes with subjects acting on behalf of that user.
FIA_USB.1.2: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [selection: [assignment: *rules for the initial association of attributes*], *"none"*].
FIA_USB.1.3: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [selection: [assignment: *rules for the changing of attributes*], *"none"*].

*Application Note:  This requirement applies CCIMB 0137.  As discussed in the CCIMB 0137, if an ST specifies any rules to apply upon initial association of attributes with subject, and any rules for allowing changing these attributes, the management of these rules should be specified in FMT.*

## 5.1.7   Class FMT: Security management

This protection profile requires support for two kinds of trusted users:  administrators and data managers.  There are a minimum of three administrators:  Security Administrator, Cryptographic Administrator, and Auditor, and there are multiple data managers.  The Security Administrator is for general security administrative responsibilities, and it's anticipated that a compliant implementation may refine and iterate this role as necessary to support component parts of the TOE, e.g., a Directory Administrator and a Platform Administrator.   Data managers are specific users granted access to a set of trusted data by a security administrator.  There may be multiple users who assume a data manager role, e.g., a CA updating directory data, and a time synchronization system.

In this protection profile the FMT_MOF family is only used to restrict the ability to enable or disable certain security functions.  All other restrictions on actions with respect to security functions are specified through FMT_MTD, because these actions all are performed through management of TSF data.

**FMT_MOF.1(1)        Management of security functions behaviour (Directory Functions)**

FMT_MOF.1.1(1) – The TSF shall restrict the ability to *enable, disable* the functions: [
- Security Alarms (FAU_ARP.1);
- Generation of evidence of origin on a per-replica-agreement basis (FCO_PRA_(EXT).1.1);
- Generation of evidence of receipt on a per-replica-agreement basis (FCO_PRA_(EXT).1.2);
- Replication of Directory Data (FDD_RPL_EXT.1);
- relying party operation replay detection mechanism (FPT_RPL.1)]

to [the Security Administrator].

*Application Note: This requirement ensures only the Security Administrator can enable or disable (turn on or turn off) the alarm notification function. As currently written, FAU_ARP.1 does not lend itself to behavior modification. If the ST author were to include additional functionality in FAU_ARP.1 (e.g., notify the administrator via a pager) then the ST author should consider using FMT_MTD for this requirement.*

*It should be noted that for items b and c, "per-replica-agreement basis" is intended to allow a TOE that has agreements to perform replication services with multiple peers to not generate the evidence for some of the peers while generating the evidence for the rest of the peers.*

**FMT_MOF.1(2)        Management of security functions behaviour (Cryptographic Module Testing)**

FMT_MOF.1.1(2) – The TSF shall restrict the ability to *enable, disable*  the functions: [
- cryptomodule testing after key generation (FPT_TST.1(1) and FPT_TST.1(2))]

to [the Cryptographic Administrator].


**FMT_MSA.1  Management of security attributes (directory access control attributes)**

FMT_MSA.1.1 – The TSF shall enforce the [Directory Access Control SFP] to restrict the ability to *change_default, query, modify, delete,* [selection: [assignment: [*other attribute operations*], *"none"*]] the security attributes [in the referenced policy] to [the Security Administrator, Data Manager].


**FMT_MTD.1(1)        Management of TSF data (Administration of Security Functions)**

FMT_MTD.1.1(1) – The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear* [assignment: *other operations*]] the [TSF data listed below, and all other TSF data except data explicitly mentioned in other iterations of FMT_MTD.1:
   a) TSF data required to manage the non-repudiation functions:
   b) Timeframe for receipt of acknowledgement (FCO_PRA_(EXT).1.4);
   c) TSF data required to manage the Identification and authentication functions:
   d) Authentication failure handling (FIA_AFL.1-NIAP-0425);
   e) Authentication Mechanisms and Rules for Authentication (FIA_UAU.*);
   f) Anonymous user access including any security administrator defined default subject security attribute for these anonymous  users (FIA_UAU.* and FIA_USB.1-NIAP-0351)

*Application Note:  Management of the distributed authentication methods requires that only a security administrator may define which methods may be allowed for trusted peer directories.*

*Regarding anonymous user access, it is expected that ST authors will provide a description of how anonymous access may be disabled in their TOE summary specification section.*

g) TSF data required to manage the session locking and session establishment functions:
h) Session Locking of local interactive session (FTA_SSL.1);
i) Session Locking of Remote Administration Session (FTA_SSL.3(1));
j) Session locking of Remote directory service session (FTA_SSL.3(2))
k) Session Establishment conditions (FTA_TSE.1)
l) TSF data required to manage the Audit and Alarm functions:
m) maintenance of the users with read access to the audit records (auditor and data manager for audit information (FAU_SAR.1(2));
n) maintenance of the rules by (adding, modifying, deletion) of rules from the set of rules (FAU_SAA.1-NIAP-0407);
o) TSF self-tests (FPT_TST_(EXT).1);
p) Automated and Manual recovery from a failure or service discontinuity (FPT_RCV.2);
q) Managing the group of users that are part of a role (FMT_SMR.2);
r) Maintenance of banner message (FTA_TAB.1);
s) Managing the replication agreements (FDD_RPL_(EXT).1);
t) Specifying the actions to be taken when the TSF data is at or exceeds the limits defined for FMT_MTD.2*]

to [the Security Administrator].

*Application Note: If multiple administrators are used to implement the security administrator role, the ST author should iterate this component refine the security administrator role assignment appropriately. The last item, TSF data maintained inside or outside Directory Information Base, is used as a catch-all to ensure access is secure. Management of sets of this data can be delegated to a data manager in FMT_MTD.1(4)*

**FMT_MTD.1(2)      Management of TSF data (cryptographic TSF data)**

FMT_MTD.1.1(2) – The TSF shall restrict the ability to *modify, query, and clear* the [cryptographic security data] to [the Cryptographic Administrator].

*Application Note: The intent of this requirement is to restrict the ability to configure the TOE's cryptographic policy to the Cryptographic Administrator. Configuring the cryptographic policy is related to things such as: setting modes of operation, key lifetimes, selecting a specific algorithm, and key length.*

**FMT_MTD.1(3)      Management of TSF data (time TSF data)**

FMT_MTD.1.1(3) – The TSF shall restrict the ability to [*set*] the [time and date used to form the time stamps in FPT_STM.1] to [the Security Administrator and Authorized Data Manager].

*Application Note: The access granted to an authorized data manager is to provide a means for a Trusted External IT entity to synchronize the TOE's time with an external time source, e.g., an external NTP server.*

*The ability to query the directory information base is not included in this requirement so relying parties can read certificates and RLs.*

**FMT_MTD.1(4)      Management of TSF data (Subsets of TSF data)**

FMT_MTD.1.1(4) – **Refinement:** The TSF shall restrict the ability to [selection: *create, query, modify, delete, clear*, [selection: [assignment: *other operations*], *"none"*]] [sets of TSF data defined by a security administrator] to [a data manager].

*Application Note: The intent of this requirement is to allow the security administrator to define a sub-hierarchy of the directory to which a data manager has (essentially) administrative access. This can include creating a sub-hierarchy, modifying a sub-hierarchy, and having access to certain information (e.g., certificate-related data) in this sub-hierarchy. ST authors should iterate or refine this requirement to reflect the capabilities of the particular TOE.*

**FMT_MTD.2(1)        Management of limits on TSF data (processor time percentage)**

FMT_MTD.2.1(1) – The TSF shall restrict the specification of the limits for [the percentage of processor time used by a relying party, and the time period over which this percentage is calculated] to [the Security Administrator].
FMT_MTD.2.2(1) – The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [assignment: *actions to be taken*].

*Application Note: The ST author should specify the actions that the TOE takes when quota is reached. For example, if the processor time is being consumed for a very large search on behalf of the relying party, the search may be terminated by the TSF. This requirement applies to the quotas specified by FRU_RSA.1(1). Note that if these actions are configurable by the administrator, the ST author should modify the audit requirements because of the CC Audit note for FMT_MTD.2.2 at the basic level.*

**FMT_MTD.2(2)        Management of limits on TSF data (transport-layer quotas)**

FMT_MTD.2.1(2) – The TSF shall restrict the specification of the limits for [quotas on transport-layer connections] to [the Security Administrator].
FMT_MTD.2.2(2) – The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [assignment: *actions to be taken*].

*Application Note: The ST author should specify the actions that the TOE takes when quota is reached. For the TCP SYN attack, for example, the action might be to drop the oldest "n" half-open connections. Note that if these actions are configurable by the administrator, the ST author should modify the audit requirements because of the CC Audit note for FMT_MTD.2.2 is at the basic level.*

**FMT_SMF.1 Specification of Management Functions**

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [backup and recovery, and archival of audit data].

**FMT_SMR.2(1)       Restrictions on security roles (strict separation)**

FMT_SMR.2.1(1) – The TSF shall maintain the roles: [
   a) Security Administrator,
   b) Auditor,
   c) Cryptographic Administrator,
   d) [selection: [assignment: *additional authorised identified roles requiring strict separation*], *"none"*]].

*Application Note:  If multiple administrators are used to implement the 'security administrator' role, the ST author should refine the security administrator role and modify the relevant assignments appropriately ensuring the O.ADMIN_ROLES object is satisfied.*

FMT_SMR.2.2(1) – The TSF shall be able to associate users with roles.
FMT_SMR.2.3(1) – **Refinement:** The TSF shall ensure the **following** conditions are satisfied: [
   a) a user may act in only one role at a time without re-authenticating to a new role;
   b) all roles are distinct; that is, there shall be no overlap of operations performed through administrative interfaces by each role;
   c) all roles shall be able to administer the TOE locally;
   d) all roles shall be able to administer the TOE remotely; and
   e) [selection: [assignment: *additional conditions for the different roles*], *"none"*]].

*Application Notes:  In the first bullet of FMT_SMR.2.3(1), the intent is to allow a single user to fill multiple roles, but not at the same time.  Note that this means that if the TOE uses the "user group" mechanism to implement roles, they have to ensure that only one group representing a role is "active" at a time, and that changing the "active" group to a new role-representing group requires the user to re-authenticate.*

*The second bullet indicates that the functions available to the role must not overlap.  While it is true that a platform administrator may be able to indirectly affect directory functions (by directly editing a platform file containing directory policy information), this goes beyond what is required to counter the threat (see rationale section).  The intent is that the interface presented to the role (and described in the AGD_ADM documentation) is unique with respect to the presented functionality for each role.*

*In the selection for SMR.2.3(1), the ST author should fill in the assignment for any additional conditions the TOE places on the roles, or select "none" if there are no additional conditions.*

**FMT_SMR.2(2)       Restrictions on security roles (data administration and users)**

FMT_SMR.2.1(2) – The TSF shall maintain the roles: [
   a) Data Manager;
   b) Relying Party; and
   c) [selection: [assignment: *additional authorised identified roles*], *"none"*]].

*Application Note:  It's expected that multiple data managers will be used to implement the data manager role, the ST author should refine the data manager role and modify the relevant assignments  appropriately.*

FMT_SMR.2.2(2) – The TSF shall be able to associate users with roles.

FMT_SMR.2.3(2) – **Refinement:** The TSF shall ensure the **following** conditions are satisfied: [
   a)  a user may act in only one role at a time without re-authenticating to a new role;
   b)  each data manager must have a user identity associated with a security administrator-specified set of trusted data for which they have access.
   c)  data managers shall be able to access the TOE locally;
   d)  all roles shall be able to access the TOE remotely; and
   e)  [selection: [assignment: *additional conditions for the different roles*], *"none"*]].

*Application Notes:  As was the case with the first iteration of this component, in the first bullet of FMT_SMR.2.3(2) the intent is to allow a single user to fill multiple roles, but not at the same time.*

*The distinction between this iteration and the previous iteration is that this iteration does not require the functions of the two roles to be distinct.  This is because the intent is that a directory manager would be responsible for only a part of the directory hierarchy, and their access (scope of control) is determined by the directory administrator (see FMT_MTD.1(5)).  So, they would be allowed to perform some of the same functions as the directory administrator, but their scope of control would be less than the entire directory.*

*In the selection for SMR.2.3(2), the ST author should fill in the assignment for any additional conditions the TOE places on the roles, or select "none" if there are no additional conditions.*

### 5.1.8   Class FPT: Protection of the TOE Security Functions

**FPT_ITA.1   Inter-TSF availability within a defined availability metric**

FPT_ITA.1.1   The TSF shall ensure the availability of [certificates and RLs] provided to a remote trusted IT product within [an security administrator-configurable time, and at a minimum 20 seconds] given the following conditions [assignment: *conditions to ensure availability*].

**FPT_RCV.2   Recovery from failure**

FPT_RCV.2.1– When automated recovery from [selection: [assignment: *list of failures/service discontinuities*], *"no failures/service discontinuities"*],  is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2– For [selection: [assignment: *list of failures/service discontinuities*], *"no failures/service discontinuities"*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

*Application Note:  This requirement applies CCIMB 0056.*

**FPT_RPL.1   Replay detection**

FPT_RPL.1.1 – The TSF shall detect replay for the following entities [remote authentication information].

FPT_RPL.1.2 – The TSF shall perform [
   a)  reject data;
   b)  audit event; and
   c)  [assignment: *list of specific actions*]]
when replay is detected.

**FPT_STM.1   Reliable time stamps**

FPT_STM.1.1 – The TSF shall be able to provide reliable time stamps for its own use.

**FPT_TDC.1(1)        Inter-TSF basic TSF data consistency (Directory Time for certificate-based security mechanisms and non-repudiation services)**

FPT_TDC.1.1(1) – The TSF shall provide the capability to consistently interpret [time stamps used by the directory portions of the TSF] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2(1) – The TSF shall use [UTC time format] when interpreting the TSF data from another trusted IT product.

*Application Note:  Synchronized and consistent interpretation of time is required for certificate-based mechanisms to accurately process the validity time of the certificate.  The TOE requires a certificate-based mechanism for authentication (FIA_UAU.5).  The TOE also requires non-repudiation services that depend on synchronized time (FCO_PRA_(EXT). 1).*

**FPT_TDC.1(2)        Inter-TSF basic TSF data consistency (Distinguished Name Character Support)**

FPT_TDC.1.1(2) – The TSF shall provide the capability to consistently interpret [Distinguished Names used by the directory portions of the TSF] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2(2) – **Refinement:** The TSF shall **support the following list of characters**: [
   a)  upper and lower case standard English language alphabetic characters;
   b)  digits (0 - 9);
   c)  spaces; and
   d)  the following punctuation and special characters:  @ # & * ( ) - \ ; : ' " , . /]
when interpreting the TSF data from another trusted IT product.

**Explicit: TSF Testing (FPT_TST_EXP.1)**

FPT_TST_EXP.1.1 The TSF shall run a suite of self tests <u>during the initial start-up and also either periodically during normal operation,</u> or at the request of an authorized administrator to demonstrate the correct operation of the TSF.
FPT_TST_EXP.1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.
*Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .*

**TSF Testing (for cryptography) (FPT_TST.1(1))**

FPT_TST.1.1(1) **Refinement:** The TSF shall run a suite of self tests **in accordance with FIPS PUB 140-2 and Appendix A of this profile** <u>during initial start-up **(on power on)**</u>, at the request of the **cryptographic administrator (on demand)**, **under various conditions <u>defined in section 4.9.1 of FIPS 140-2,</u>** and periodically **(at least once a day)** to demonstrate the correct operation of the **following cryptographic functions:i**
          a)  **key error detection;**
          b)  **cryptographic algorithms;**
          c)  **RNG/PRNG**

*Application Note: These tests apply regardless of whether the cryptographic functionality is implemented in hardware, software, or firmware.*

FPT_TST.1.2(1) **Refinement:** The TSF shall provide authorized **cryptographic administrators** with the capability to verify the integrity of **TSF data related to the cryptography by using TSF-provided cryptographic functions**.ii

*Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services*

.FPT_TST.1.3(1) **Refinement:** The TSF shall provide authorized **cryptographic administrators** with the capability to verify the integrity of stored TSF executable code **related to the cryptography by using TSF-provided cryptographic functions**.iii

*Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .*

**TSF Testing (for key generation components) (FPT_TST.1(2))**

FPT_TST.1.1(2) **Refinement**: The TSF shall **perform** self tests **immediately after generation of a key** to demonstrate the correct operation **of each key generation component**. **If any of these tests fails, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140-2 for failing a self-test, and this event will be audited**.iv

*Application Note: Key generation components are those critical elements that compose the entire key generation process (e.g., any algorithms, any RNG/PRNGs, any key generation seeding processes, etc.).*

*Application Note: These self-tests on the key generation components can be executed here as a subset of the full suite of self-tests run on the cryptography in FPT_TST.1(1) as long as all elements of the key generation process are tested.*

FPT_TST.1.2(2) **Refinement:** The TSF shall provide authorized **cryptographic administrators** with the capability to verify the integrity of TSF data **related to the key generation by using TSF-provided cryptographic functions**.v

*Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services*

.FPT_TST.1.3(2) **Refinement:** The TSF shall provide authorized **cryptographic administrators** with the capability to verify the integrity of stored TSF executable code **related to the key generation by using TSF-provided cryptographic functions**.vi

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .

### 5.1.9  Class FRU:  Resource Utilisation

**FRU_RSA.1(1)          Maximum quotas (processor time)**

FRU_RSA.1.1(1) – **Refinement:**  The TSF shall enforce **Security Administrator-specified** maximum quotas of the following resources:  [processor time] that *a Relying Party and* **[selection: [assignment: *group of users*], *"none"*]** can use *over a specified period of time*.

**FRU_RSA.1(2)          Maximum quotas (transport–layer)**

FRU_RSA.1.1(2) – Refinement:  The TSF shall enforce **Security Administrator-specified** maximum quotas of the following resources: [transport-layer representation] that *individual users* can use *simultaneously*.

*Application Note: "Transport-layer representation" refers specifically to the TCP SYN attack, where half-open connections are established thus exhausting the connection table resource.  If the TOE does not implement the*

*TCP/IP protocol, this requirement would apply to a similar type of transport-layer entity for that TOE's protocol stack.*

## 5.1.10 Class FTA: TOE Access

**FTA_SSL.1     TSF-initiated session locking**

FTA_SSL.1.1 – **Refinement:**  The TSF shall lock a **local** interactive session after [a Security Administrator-specified time period of inactivity] by:
   a)  clearing or overwriting display devices, making the current contents unreadable;
   b)  disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 – **Refinement:** The TSF shall require the **user to re-authenticate** prior to unlocking the session.

*Application Note:  A configurable expiry time for the bind token is an example implementation for this requirement.*

**FTA_SSL.2     User-initiated locking**

FTA_SSL.2.1 – **Refinement:** The TSF shall allow user-initiated locking of the user's own **local** interactive session by:
   a)  clearing or overwriting display devices, making the current contents unreadable;
   b)  disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 – **Refinement:** The TSF shall require the **user to re-authenticate** prior to unlocking the session.

*Application Note: The interactive sessions in FTA_SSL.1 and FTA_SSL.2 are those of the local administrator. Non-administrators only have remote access to the TOE and the requirements for session locking levied on them are specified in FTA_SSL.3.*

**FTA_SSL.3(1) TSF-initiated termination (remote administration session)**

FTA_SSL.3.1(1) – **Refinement:** The TSF shall terminate **a remote administration** session after a [Security Administrator-configurable time interval of session inactivity].

*Application Note: Remote administration sessions include all access by the administrators and the trusted external IT entities granted access by the security administrator.*

**FTA_SSL.3(2) TSF-initiated termination (remote directory service session)**

FTA_SSL.3.1(2) – **Refinement:** The TSF shall terminate **a remote directory services** session after a [Security Administrator-configurable time interval of session inactivity].

*Application Note: Remote directory service sessions include all access by relying parties, and Data Managers (users and trusted external IT entities) authorized by the Security Administrator to manage directory data.  This component is listed separately from the remote administration iteration to require separate control for the different types of sessions.*

**FTA_TAB.1    Default TOE access banners**

FTA_TAB.1.1 – **Refinement:** Before establishing an **administrative** session, the TSF shall display **only a Security Administrator-specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.


*Application Note: The access banner applies only when an administrator begins an interactive session with the TOE. The intent of this requirement is to advise users of warnings regarding the unauthorized use of the TOE and to provide the Security Administrator with control over what is displayed (e.g., if the Security Administrator chooses, they can remove banner information that informs administrators of the product and version number).*

**FTA_TSE.1    TOE session establishment**

FTA_TSE.1.1 – The TSF shall be able to deny session establishment based on [location, time, and day].

*Application Note: "Location" can refer to the network domain that the user (e.g., relying party) originates from.  It should be noted that this requirement applies to both relying parties and administrators of the TSF.  Also note that there may be two types of "sessions" for a TOE: one type for administration (e.g., a security administrator "logs on" to the platform, thus establishing a session) and one type for directory services, (e.g., a directory manager or a relying party binds to the directory, thus establishing a session).*

## 5.1.11 Class FTP:  Trusted path/channels

The requirements in this class are extended to remove a contradiction in the original requirements, FTP_ITC.1 and FTP_TRP.1.  Based on OD-232 an interpretation is being created by NIAP to fix the contradictory wording.  When the NIAP interpretation is created and the international community adopts the new wording as a final interpretation, compliant TOEs should use the updated requirements rather than the extended requirements.

The trusted channel and trusted path requirements are specified using two iterations for each.  The iterations for each apply to the same set of operations but specify requirements for different aspects of the trusted channel or trusted path, i.e., 1.  prevention of disclosure, and 2. detection of modification.  A compliant TOE provides a trusted channel and trusted path for these operations that provides the services specified by both iterations.

**Extended:  Inter-TSF trusted channel (Prevention of Disclosure)  FTP_ITC_(EXT).1(1)**

FTP_ITC_(EXT).1.1(1) –The TSF shall use encryption to provide a trusted communication channel between itself and a trusted external IT entity that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure.

*Application Note: Since a symmetric algorithm is required, the symmetric key will either have to be generated (FCS_CKM.1) or otherwise established (FCS_CKM_(EXT).2).  The ST may wish to include an application note indicating what mechanism(s) are used for keying the algorithm used to provide the above functionality.*


FTP_ITC_(EXT).1.2(1) – The TSF shall permit the TSF, or the trusted external IT entity to initiate communication via the trusted channel.

*Application Note: The encryption used to protect the communication channel from disclosure is the symmetric algorithm specified in FCS_COP_(EXT).2.*

*The encryption used to protect the communication channel from disclosure can encryption/decryption specified in FCS_COP_(EXT).2, ensuring the strength of the mechanism is commensurate with medium robustness requirements. If an implementation uses another cryptographic algorithm, it's expected that the ST author will include this algorithm as another FCS_COP requirement with enough information to enable a comparison of its strength and applicability to support this security function.*

*FTP_ITC_(EXT).1.2(1) is used to ensure secure communications between the TOE and an external trusted IT entity (e.g., Peer TOE, Peer Directory, time synchronization system). While these trusted IT entities may initiate communications, it may be the case that the TOE is required to perform a "pull" operation (e.g., obtaining time from a time server).*

FTP_ITC_(EXT).1.3(1) – The  trusted channel shall be used for all password-based authentication functions, replication operations, remote management of directory service data, and [selection: [assignment: *list of other functions for which a trusted channel is required*], *"none"*].

*Application Note: The "other functions" are the services that are provided by the trusted IT entities (e.g., time server, intrusion detection system access). If the ST author wishes to specify the function for which trusted channel is initiated by the TSF vs. the trusted IT entities, then this requirement should be iterated.*

**Extended:  Inter-TSF trusted channel (Detection of Modification) FTP_ITC_(EXT).1(2)**

FTP_ITC_(EXT).1.1(2) – The TSF shall use a cryptographic signature to provide a trusted communication channel between itself and trusted external IT entity that is logically distinct from other communication channels and provides assured identification of its end points and detection of the modification of data.
FTP_ITC_(EXT).1.2(2) – The TSF shall permit the TSF, or the trusted external IT entity to initiate communication via the trusted channel.

*Application Note: The encryption used to detect modification in a communication channel can be a digital signature/verification algorithm specified in FCS_COP_(EXT).3, ensuring the strength of the mechanism is commensurate with medium robustness requirements. If an implementation uses another cryptographic algorithm, it's expected that the ST author will include this algorithm as another FCS_COP requirement with enough information to enable a comparison of its strength and applicability to support this security function.*

*FTP_ITC_(EXT).1.2(2) is used to ensure secure communications between the TOE and an external trusted IT entity (e.g., Peer TOE, Peer Directory, time synchronization system). While these trusted IT entities may initiate communications, it may be the case that the TOE is required to perform a "pull" operation (e.g., obtaining time from a time server).*

FTP_ITC_(EXT).1.3(2) – The  trusted channel shall be used for all password-based authentication functions, replication operations, remote management of directory service data, and [selection: [assignment: *list of other functions for which a trusted channel is required*], *"none"*].

*Application Note: The "other functions" are the services that are provided by the trusted IT entities (e.g., time server, intrusion detection system access). If the ST author wishes to specify the function for which trusted channel is initiated by the TSF vs. the trusted IT entities, then this requirement should be iterated.*

**Extended:  Trusted path (Prevention of Disclosure)  FTP_TRP_(EXT).1(1)**

FTP_TRP_(EXT).1.1(1) – The TSF shall provide an encrypted communication path between itself and a remote administrator or relying party authenticating with a password that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure.

FTP_TRP_(EXT).1.2(1) – The TSF shall permit a remote administrator or relying party authenticating with a password to initiate communication via the trusted path.

FTP_TRP_(EXT).1.3(1) – The TSF shall require the use of the trusted path for relying party password-based authentication, all remote administration actions, [selection: [assignment: *other services for which trusted path is required*], *"none"*].

*Application Note: The encryption used to protect the communication channel from disclosure can use encryption/decryption specified in FCS_COP_(EXT).2, ensuring the strength of the mechanism is commensurate with medium robustness requirements. If an implementation uses another cryptographic algorithm, it's expected that the ST author will include this algorithm as another FCS_COP requirement with enough information to enable a comparison of its strength and applicability to support this security function.*

*"all remote administration actions" means that the entire remote administration session is protected with the trusted path; that is, the administrator is assured of communicating with the TOE and the data passing between the administrator and the TOE are protected from disclosure.*


**Extended:  Trusted path (Detection of Modification)  FTP_TRP_(EXT).1(2)**

FTP_TRP_(EXT).1.1(2) –The TSF shall use a cryptographic signature to provide a trusted communication path between itself and a remote administrator and relying party authenticating with a password that is logically distinct from other communication paths and provides assured identification of its end points and detection of the modification of data.

FTP_TRP_(EXT).1.2(2) – The TSF shall permit a remote administrator and relying party authenticating with a password  to initiate communication via the trusted path.

FTP_TRP_(EXT).1.3(2) – The TSF shall require the use of the trusted path for relying party password-based authentication, all remote administration actions, [selection: [assignment: *other services for which trusted path is required*], *"none"*].

*Application Note: The encryption used to detect modification in a communication channel can be a digital signature/verification algorithm specified in FCS_COP_(EXT).3, ensuring the strength of the mechanism is commensurate with medium robustness requirements. If an implementation uses another cryptographic algorithm, it's expected that the ST author will include this algorithm as another FCS_COP requirement with enough information to enable a comparison of its strength and applicability to support this security function.*

*"all remote administration actions" means that the entire remote administration session is protected with the trusted path; that is, the administrator is assured of communicating with the TOE and the TOE provides a means for detecting the modification of data that flows through the protected communication path.*

## 5.2   SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

This Protection Profile provides functional requirements for the IT Environment. The IT environment includes trusted external IT entities (e.g., peer trusted directories, time synchronization server) and any IT entities that are used by administrators to remotely administer the TOE. These requirements consist of functional components from Part 2 of the CC.

*Application Note:  Note that for the following elements, "refinement" is indicated even though these are extended requirements in order to show that they are the "IT Environment Half" of the requirements specified for the TOE in Section .1.*

**Extended:  Inter-TSF trusted channel (Prevention of Disclosure) FTP_ITC_(EXT).1(3)**

FTP_ITC_(EXT).1.1(3) – **Refinement:** The **IT Environment** shall use encryption to provide a trusted communication channel between itself and the TSF that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure.

FTP_ITC.1.2(3) – **Refinement:** The **IT Environment** shall permit the TSF or the IT environment to initiate communication via the trusted channel.

FTP_ITC.1.3(3) – The trusted channel shall be used for all password-based authentication functions, replication operations, remote management of directory service data, and [selection: [assignment: *list of other functions for which a trusted channel is required*], *"none"*].

**Extended:  Inter-TSF trusted channel (Detection of Modification)  FTP_ITC_(EXT).1(4)**

FTP_ITC_(EXT).1.1(4) – **Refinement:** The **IT Environment** shall use a cryptographic signature to provide a trusted communication channel between itself and the TSF that is logically distinct from other communication channels and provides assured identification of its end points and detection of the modification of data.

FTP_ITC_(EXT).1.2(4) – **Refinement:** The **IT Environment** shall permit the TSF, or the IT Environment to initiate communication via the trusted channel.

FTP_ITC_(EXT).1.3(4) – The trusted channel shall be used for password-based authentication functions, replication operations, remote management of directory service data, and [selection: [assignment: *list of other functions for which a trusted channel is required*], *"none"*].

*Application Note:  The FTP_ITC_(EXT).1(*) requirements are levied on the IT environment to ensure that the necessary support exists in the IT environment to communicate securely with the TOE. The FCS family of requirements have not been explicitly stated in the IT environment requirements, since the cryptographic algorithms and key sizes are implicitly required by the IT environment in order to communicate with the TOE.*

**Extended:  Trusted path (Prevention of Disclosure)  FTP_TRP_EXP.1(3)**

FTP_TRP_(EXT).1.1(3) – **Refinement:** The **IT Environment** shall provide an encrypted communication path between itself and the TSF that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure.

FTP_TRP_(EXT).1.2(3) – **Refinement:** The **IT Environment** shall permit remote users of the TSF to initiate communication to the TSF via the trusted path.

FTP_TRP_(EXT).1.3(3) – **Refinement:**  The **IT Environment** shall initiate the use of the trusted path for relying party password-based authentication, all remote administration actions, [selection: [assignment: *other services for which trusted path is required*] *"none"*].

*Application Note:  This requirement is levied on the IT environment to ensure that the necessary support exists in the IT environment to communicate securely with the TOE. The FCS family of requirements have not been explicitly stated in the IT environment requirements, since the cryptographic algorithms and key sizes are implicitly required by the IT environment in order to communicate with the TOE.*

**Extended:  Trusted path (Detection of Modification)  FTP_TRP_(EXT).1(4)**

FTP_TRP_(EXT).1.1(4) – Refinement: The IT Environment shall use a cryptographic signature to provide a trusted communication path between itself and the TSF that is logically distinct from other communication paths and provides assured identification of its end points and detection of the modification of data.

FTP_TRP_(EXT).1.2(4) – Refinement:  The IT Environment shall permit remote users of the TSF to initiate communication to the TSF via the trusted path.

FTP_TRP_(EXT).1.3(4) – Refinement:  The IT Environment shall initiate the use of the trusted path for relying party password-based authentication, all remote administration actions, [selection: [assignment: *other services for which trusted path is required*] *"none"*].

*Application Note:  The FTP_TRP_(EXT).1(*) requirements are levied on the IT environment to ensure that the necessary support exists in the IT environment to communicate securely with the TOE. The FCS family of requirements have not been explicitly stated in the IT environment requirements, since the cryptographic algorithms and key sizes are implicitly required by the IT environment in order to communicate with the TOE.*

The following extended requirement for non-repudiation of replication activity include functions to support the Directories' role in a PKI to provide non-repudiation for the replication process required

by FDD_RPL_(EXT).1.  The non-repudiation service verifies the replication process was successful by generating evidence that the replica data was sent and received without error, as well as provide proof of the originator and recipient of the replicated data.  The requirement includes both the generation and verification of evidence for non-repudiation, including timestamps for when the replica data was sent and when its been received (as reported by the consumer), and notification when evidence of receipt the TOE is waiting for is overdue.

**Extended: Proof of Replication Activity (FCO_PRA_(EXT).1(2))**

*Application Note:  Technically the only thing that the TOE relies on the IT Environment to enforce its policies for the following requirements is that the IT Environment provide the evidence of receipt; again, this component is kept intact to re-enforce the peering relationship needed to implement this requirement.*

FCO_PRA_(EXT).1.1(2) – Refinement: The IT Environment shall be able to generate evidence of origin for transmitted Replica Data that consists of the identity of the IT Entity originating the activity, the fact that replication activity was initiated, the time that the activity was initiated, and [assignment: other information included bound as "evidence of origin"].

*Application Note:  This applies when the IT entity in the IT Environment  is the originator of the replication activity (defined in FDD_RPL_(EXT).1).  The intent is that evidence be produced that will prove that the IT entity originated a replication event at a certain time.  The assignment should be used to specify any other information that will be included (and presumably signed by the IT entity) as evidence of the initiation of a replication event activity (for instance, the name of the sub-hierarchy to be replicated).*

FCO_PRA_(EXT).1.2(2) – Refinement: The IT Environment shall be able to generate evidence of receipt for Replica Data received from the TOE that consists of the identity of the IT Entity receiving the activity, the fact that the replication data were received, the time of receipt, and [assignment: *other information included bound as "evidence of receipt"*].

*Application Note:  This applies when the IT entity in the IT Environment  is the receiver of the replication activity (defined in FDD_RPL_(EXT).1).  The intent is that evidence be produced that will prove that the IT entity  received replication data at a certain time.  The assignment should be used to specify any other information that will be included (and presumably signed by the IT entity) as evidence of the receipt of a replication event activity (for instance, the name of the sub-hierarchy to be replicated).*

FCO_PRA_(EXT).1.3(2) – **Refinement:** The **IT Environment** shall produce and maintain "evidence of replication activity" that binds, in a way that cannot be repudiated, the evidence of origin and the evidence of receipt to all fields of the replica data.

*Application Note:  For non-repudiation of replication data the requirement to relate the identity to all the fields can be satisfied using replication agreement configuration information and similar bulk loading specifications.  The intent of this requirement is to provide non-repudiation that the replication process was received, processed, and completed without error, and that the evidence used is not ephemeral.  It should be noted that in order to meet this requirement the source of the replica data will have to have some means to accept the evidence of receipt from the consumer.*

FCO_PRA_(EXT).1.4(2) – **Refinement:** When originating a replication activity, the **IT Environment** shall be able to send notification using [assignment: *mechanism(s)*] to a Security Administrator if it does not receive evidence of receipt of transmitted Replica Data within a Security Administrator-specified time period.

*Application Note: The assignment should be filled in with the mechanism or mechanisms used to send the notification to the Security Administrator; this could be e-mail, a message to the console, etc.*

FCO_PRA_(EXT).1.5(2) – **Refinement:** The **IT Environment** shall provide a capability for a Security Administrator and [selection: [assignment: *other roles*], "*no other roles*"] to verify the evidence of replication activity.

*Application Note: The assignment should be filled in with the roles that are authorized to perform the actions required to verify information about a replication event. The "evidence of replication activity" is specified in FCO_PRA_(EXT).1.3.*

## 5.3  TOE SECURITY ASSURANCE REQUIREMENTS

The TOE assurance requirements for this PP are the Medium Robustness Assurance Package and do not map to a CC EAL.  The assurance requirements are summarized in the Table 5.3 below.

**Table 5.3 – Assurance Requirements**

| Assurance Class | ASSURANCE COMPONENTS | ASSURANCE COMPONENTS DESCRIPTION |
|---|---|---|
| DEVELOPMENT | ADV_ARC.1 | Security Architectural Description |
| | **ADV_FSP.5** | **Complete semi-formal functional specification with additional error information** |
| | ADV_IMP.1 | Implementation of the TSF |
| | **ADV_INT.3** | **Minimally complex internals** |
| | **ADV_TDS.4** | **Semiformal modular design** |
| GUIDANCE DOCUMENTS | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative User guidance |
| LIFE CYCLE SUPPORT | ALC_CMC.4 | Product support, acceptance procedures and automation |
| | ALC_CMS.4 | Problem tracking CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS.1 | Identification of security measures |
| | **ALC_FLR.2** | **Flaw Reporting Procedures** |
| | ALC_LCD.1 | Developer defined life-cycle model |
| | ALC_TAT.1 | Well-defined development tools |
| TESTS | ATE_COV.2 | Analysis of coverage |
| | **ATE_DPT.3** | **Testing: modular design** |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |

| Assurance Class | ASSURANCE COMPONENTS | ASSURANCE COMPONENTS DESCRIPTION |
|---|---|---|
| VULNERABILITY ASSESSMENT | **AVA_CCA_(EXT).1** | **Systematic cryptographic module covert channel analysis (required when Cryptography is invoked)** |
| | **AVA_VAN.4** | **Methodical vulnerability analysis** |

### 5.3.1   Class ADV: Development

### 5.3.1.1 ADV_ARC.1 Security architecture description

Dependencies: ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

Evaluator action elements:

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

### 5.3.1.2 ADV_FSP.5 Complete semi-formal functional specification with additional error information

Dependencies: ADV_TDS.1 Basic design,
ADV_IMP.1 Implementation representation of the TSF Developer action elements:

Developer action elements:

ADV_FSP.5.1D The developer shall provide a functional specification.

ADV_FSP.5.2D The developer shall provide a tracing from the functional specification to the SFRs.
Content and presentation elements:

ADV_FSP.5.1C The functional specification shall completely represent the TSF.

ADV_FSP.5.2C The functional specification shall describe the TSFI using a semi-formal style.

ADV_FSP.5.3C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.5.4C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.5.5C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.5.6C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.5.7C The functional specification shall describe all error messages that do not result from an invocation of a TSFI.

ADV_FSP.5.8C The functional specification shall provide a rationale for each error message contained in the TSF implementation yet does not result from an invocation of a TSFI.

ADV_FSP.5.9C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
Evaluator action elements:

Evaluator action elements:

ADV_FSP.5.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.5.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

### 5.3.1.3 ADV_IMP.1 Implementation representation of the TSF

Dependencies: ADV_TDS.3 Basic modular design
ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

Content and presentation elements:

ADV_IMP.1.1C The implementation representation shall 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 in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

Evaluator action elements:

ADV_IMP.1.1E The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

### 5.3.1.4 ADV_INT.3 Minimally complex internals

Dependencies: ADV_IMP.1 Implementation representation of the TSF
ADV_TDS.3 Basic modular design
ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_INT.3.1D The developer shall design and implement the entire TSF such that it has well-structured internals.

ADV_INT.3.2D The developer shall provide an internals description and justification.

Content and presentation elements:

ADV_INT.3.1C The justification shall describe the characteristics used to judge the meaning of "well-structured" and "complex".

ADV_INT.3.2C The TSF internals description shall demonstrate that the entire TSF is well-structured.

Evaluator action elements:

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

ADV_INT.3.2E The evaluator *shall perform* an internals analysis on the entire TSF.

### 5.3.1.5 ADV_TDS.4 Semiformal modular design

Dependencies: ADV_FSP.5 Complete semi-formal functional specification with additional error information Developer action elements:

Developer action elements:

ADV_TDS.4.1D The developer shall provide the design of the TOE.

ADV_TDS.4.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design. Content and presentation elements:

Content and presentation elements:

ADV_TDS.4.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.4.2C The design shall describe the TSF in terms of modules, designating each module as SFR-enforcing, SFR-supporting, or SFR-non-interfering.

ADV_TDS.4.3C The design shall identify all subsystems of the TSF.

ADV_TDS.4.4C The design shall provide a semiformal description of each subsystem of the TSF, supported by informal, explanatory text where appropriate.

ADV_TDS.4.5C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.4.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.4.7C The design shall describe each SFR-enforcing and SFR-supporting module in terms of its purpose and interaction with other modules.

ADV_TDS.4.8C The design shall describe each SFR-enforcing and SFR-supporting module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.4.9C The design shall describe each SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.4.10C The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it. Evaluator action elements:

Evaluator action elements:

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

ADV_TDS.4.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

## 5.3.2  Class AGD: Guidance documents

### 5.3.2.1 AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance 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.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

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

### 5.3.2.2 AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

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

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

### 5.3.3   Class ALC: Life-cycle support

### 5.3.3.1 ALC_CMC.4 Production support, acceptance procedures and automation

Dependencies: ALC_CMS.1 TOE CM coverage
ALC_DVS.1 Identification of security measures
ALC_LCD.1 Developer defined life-cycle model

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

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.4.1C The TOE shall be labeled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorized changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

Evaluator action elements:

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

## 5.3.3.2 ALC_CMS.4 Problem tracking CM coverage

Dependencies: No dependencies.

Developer action elements:

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.4.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

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

## 5.3.3.3 ALC_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

Evaluator action elements:

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

### 5.3.3.4 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 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.

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.

### 5.3.3.5 ALC_FLR.2 Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.2.1D The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation elements:

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

Evaluator action elements:

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

### 5.3.3.6 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 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.

### 5.3.3.7 ALC_TAT.1 Well-defined development tools

Dependencies: ADV_IMP.1 Implementation representation of the TSF

Developer action elements:

ALC_TAT.1.1D The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of each development tool.

Content and presentation elements:

ALC_TAT.1.1C Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3C The documentation of each development tool 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.

### 5.3.4  Class ATE: Tests

### 5.3.4.1 ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.2 Security-enforcing 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 elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

Evaluator action elements:

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

### 5.3.4.2 ATE_DPT.3 Testing: modular design

Dependencies: ADV_ARC.1 Security architecture description
ADV_TDS.4 Semiformal modular design
ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.3.1D The developer shall provide the analysis of the depth of testing.

Content and presentation elements:

ATE_DPT.3.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design.

ATE_DPT.3.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.3.3C The analysis of the depth of testing shall demonstrate that all TSF modules in the TOE design have been tested.

Evaluator action elements:

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

## 5.3.4.3 ATE_FUN.1 Functional testing

Dependencies: ATE_COV.1 Evidence of coverage

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 elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator action elements:

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

## 5.3.4.4 ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.2 Security-enforcing functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures
ATE_COV.1 Evidence of coverage
ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation 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 execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

## 5.3.5  Class AVA: Vulnerability assessment

### 5.3.5.1 AVA_CCA_(EXT).1 Systematic Cryptographic Module covert channel analysis

Dependencies:  ADV_FSP.4 Complete Functional Specification

ADV_IMP.1 Implementation of the TSF

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative User guidance

Application notes: The covert channel analysis is performed only upon the cryptographic module; a search is made for the leakage of critical cryptographic security parameters from the cryptographic module, rather than a violation of an information control policy. Inappropriate handling / leakage of any critical cryptographic security parameters (covered or not) that by design and implementation lie outside the cryptographic module is not addressed by this CCA. Thus, leakage of such parameters in such designs and implementations must be investigated by other means.

Developer action elements:

AVA_CCA_(EXT).1.1D  For the cryptographic module, the developer shall conduct a search for covert channels for the leakage of critical cryptographic security parameters whose disclosure would compromise the security provided by the module.

Application Note: The remainder of the TOE need not be subjected to a covert channel analysis. (Ideally, a covert channel analysis on the entire TSF would determine if TSF interfaces can be used covertly for the leakage of critical cryptographic security parameters. While such extensive covert channel analysis is more complete, it is also difficult and expensive. At this time it is considered beyond the scope of effort and cost considered reasonable for COTS medium robustness products. Consequently, covert channel analysis has been limited here to the cryptographic module, but that analysis limitation does come with some added risk of unknown leakage from other parts of the TOE.

AVA_CCA_(EXT).1.2D  The developer shall provide covert channel analysis documentation.

Content and presentation of evidence elements:

AVA_CCA_(EXT).1.1C  The analysis documentation shall identify covert channels in the cryptographic module and estimate their capacity.

AVA_CCA_(EXT).1.2C  The analysis documentation shall describe the procedures used for determining the existence of covert channels in the cryptographic module, and the information needed to carry out the covert channel analysis.

AVA_CCA_(EXT).1.3C  The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA_CCA_(EXT).1.4C  The analysis documentation shall describe the method used for estimating channel capacity, based on worst-case scenarios.

AVA_CCA_(EXT).1.5C  The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA_CCA_(EXT).1.6C  The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

Evaluator action elements:

AVA_CCA_(EXT).1.1E  The NSA evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_CCA_(EXT).1.2E  The NSA evaluator shall confirm that the results of the covert channel analysis show that the cryptographic module meets its functional requirements.

AVA_CCA_(EXT).1.3E  The NSA evaluator shall selectively validate the covert channel analysis through independent analysis and testing.

Application Note: The cryptographic security parameters are to be defined in the Security Target


## 5.3.5.2 AVA_VAN.4 Methodical vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description
ADV_FSP.2 Security-enforcing functional specification
ADV_TDS.3 Basic modular design
ADV_IMP.1 Implementation representation of the TSF
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures


Developer action elements:

AVA_VAN.4.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.4.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

AVA_VAN.4.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.4.3E The evaluator shall perform an independent, methodical vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.4.4E The evaluator shall conduct penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing Moderate attack potential.

# 6 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4 and Section 5, respectively. Additionally, this section describes the rationale for not satisfying all of the dependencies.

## *6.1 RATIONALE FOR TOE SECURITY OBJECTIVES*

### Table 6.1 – Security Objectives to Threats and Policies Mappings

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| T.ADMIN_ERROR<br><br>An administrator may incorrectly install or configure the TOE, or install a corrupted TOE, resulting in ineffective security mechanisms. | O.ROBUST_ADMIN_GUIDANCE<br><br>The TOE will provide administrators with the necessary information for secure delivery and management.<br><br>O.ADMIN_ROLE<br><br>The TOE will provide administrator roles to isolate administrative actions.<br><br>O.MANAGE<br><br>The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. | O.ROBUST_ADMIN_GUIDANCE (ALC_DEL.1, AGD_PRE.1, AGD_OPE.1) help to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.<br><br>O.ADMIN_ROLE (FMT_SMR.2(1)-(2)) plays a role in mitigating this threat by limiting the functions an administrator can perform in a given role. For example, the Audit Administrator could not make a configuration mistake that would impact the directory access control policy. Likewise, a directory manager could only affect policies in the sub-hierarchy they are responsible for, and not other sub-hierarchies or global directory policies.<br><br>O.MANAGE (FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(4) FMT_SMF.1) also contributes to mitigating this threat by providing administrators the capability to view configuration settings. For example, if the Directory Administrator made a mistake when configuring the directory schema, providing them the capability to view and manipulate the schema affords them the ability to discover any mistakes that might have been made. In addition administrators have the capability to recover from an error or corrupted TSF data. |
| T.ADMIN_ROGUE<br><br>An administrator's intentions may become malicious resulting in user or TSF data being compromised. | O.ADMIN_ROLE<br><br>The TOE will provide administrator roles to isolate administrative actions. | O.ADMIN_ROLE (FMT_SMR.2(1)) mitigates this threat by restricting the functions available to an administrator. Note that this restriction is not strict, since the isolation provided by the component only applies to functions available to the role through an interface. So, for example, if there was an administrative interface to review the audit data, then that interface would have to be restricted to the Auditor. An 'all-powerful' role, e.g., root, also needs to be restricted by the TSF (and not by policy or guidance). Further, suppose that the security administrator had an interface that available that could read any memory location. While it is a fact that the security administrator could use the memory reader interface to read the audit file while the auditor is examining it, this type of separation is not what is required by FMT_SMR.2(1). Consequently, mitigation of T.ADMIN_ROUTE is somewhat less than what would be achieved through total isolation of roles, but is somewhat more than would be |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | achieved through no separation at all.<br><br>The mitigation provided by the objective to this threat is somewhat different than the part this objective plays in countering T.ADMIN_ERROR, in that this presumes that separate individuals will be assigned separate roles. If the Audit Administrator's intentions become malicious they would not be able to render the TOE unable to enforce its directory access control policy. On the other hand, if the Directory Administrator becomes malicious they could affect the directory access control policy, but the Audit Administrator may be able to detect those actions. |
| **T.AUDIT_COMPROMISE**<br><br>A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. | O.AUDIT_PROTECTION<br><br>The TOE will provide the capability to protect audit information.<br><br>O.RESIDUAL_INFORMATION<br><br>The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.<br><br>O.SELF_PROTECTION<br><br>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. | O.AUDIT_PROTECTION (FAU.SAR.2, FAU_STG.1-NIAP-0429, FAU_STG.3, FAU_STG.NIAP-0414-1-0429-1, FMT_SMF.1) contributes to mitigating this threat by controlling access to the audit trail. The auditor and any trusted IT entities performing IDS-like functions are the only ones allowed to read the audit trail. No one is allowed to modify audit records, and the Auditor is the only one allowed to delete audit records in the audit trail. The TOE has the capability to prevent auditable actions from occurring if the audit trail is full, and of notifying an administrator if the audit trail is approaching its capacity. In addition, the TOE has the capability to restore audit data corrupted by the attacker.<br><br>O.RESIDUAL_INFORMATION (FDP.RIP.2) prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data.<br><br>O.SELF_PROTECTION ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation. |
| **T.CRYPTO_COMPROMISE**<br><br>A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromise the cryptographic mechanisms and the data protected by those mechanisms. | O.CRYPTOGRAPHY_VALIDATED<br><br>The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions.<br><br>O.CRYPTOGRAPHIC_FUNCTIONS<br><br>The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.<br><br>O.SELF_PROTECTION<br><br>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. | O.CRYPTOGRAPHY_VALIDATED (FCS_BCM_(EXT).1) contributes to mitigating this threat by requiring FIPS-approved functions to be used, thus lessening the chance that a poorly-thought-out algorithm could be compromised by an adversary. Additionally, the requirements levied on the cryptomodule by the FIPS process, and the verification of those requirements by the FIPS labs, helps add assurance that the cryptographic module can protect itself.<br><br>O.CRYPTOGRAPHIC_FUNCTIONS (FCS_CKM.4 and FCS_CKM_(EXT).2) mitigates the possibility of malicious users or processes from gaining inappropriate access to cryptographic data, including keys. This objective ensures that the cryptographic data does not reside in a resource that has been used by the cryptographic functions and then reallocated to |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | O.DOCUMENT_KEY_LEAKAGE<br><br>The bandwidth of channels that can be used to compromise key material shall be documented. | another process.<br><br>O.SELF_PROTECTION ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation. .<br><br>O.DOCUMENT_KEY_LEAKAGE (AVA_CCA_(EXT).2) addresses this threat by requiring the developer to perform an analysis that documents the amount of key information that can be leaked via a covert channel. This provides information that identifies how much material could be inappropriately obtained within a specified time period. |
| T.MASQUERADE<br><br>A user may masquerade as another entity in order to gain access to data or TOE resources. | O.ROBUST_TOE_ACCESS<br><br>The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate<br><br>O.TRUSTED_PATH<br><br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. | O.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_ATD.1(1)-(3), FIA_UID.2, FIA_UAU.1, FIA_UAU.2, FIA_UAU.5, FTA_TSE.1, , FPT_TDC.1(1)-(2), FPT_ITA.1) mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanisms, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE. This objective also allows the TOE to correctly interpret information used during the authentication process so that it can make the correct decisions when identifying and authenticating users. Finally, this objective provides the ability to control access to certificates and revocation lists so they are available in a timely fashion, contributing to correct authentication decisions.<br><br>O.TRUSTED_PATH (FTP_ITC_(EXT).1(1), FTP_ITC_(EXT).1(2)) ensures that the communication path end points between the TOE and trusted IT entities are defined. This mechanism allows the TOE to be assured that it is communicating with a trusted IT entity, and that another (untrusted) entity is not attempt to access TSF resources. |
| T.FLAWED_DESIGN<br><br>Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program. | O.CHANGE_MANAGEMENT<br><br>The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.<br><br><br>O.SOUND_DESIGN<br><br>The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented.<br><br>O.VULNERABILITY_ANALYSIS_ TEST | O.SOUND_DESIGN (ADV_FSP.5, ADV_TDS.4, ADV_INT_.1,) counters this threat, to a degree, by requiring that the TOE be developed using sound engineering principles. By accurately and completely documenting the design of the security mechanisms in the TOE, including a security model, the design of the TOE can be better understood, which increases the chances that design errors will be discovered.<br><br>O.CHANGE_MANAGEMENT ((ALC_CMC.4, ALC_CMS.4, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1) plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. This includes controlling physical access to the TOE's development area, and having an automated configuration |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. | management system that ensures changes made to the TOE go through an approval process and only those persons that are authorized can make changes to the TOE's design and its documentation. |
| | | O.VULNERABILITY_ANALYSIS_TEST (AVA_VAN.4) ensures that the design of the TOE is independently analyzed for design flaws. Having an independent party perform the assessment ensures an objective approach is taken and may find errors in the design that would be left undiscovered by developers that have a preconceived incorrect understanding of the TOE's design. |
| T.CORRUPTED_IMPLEMENTATION  Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program. | O.CHANGE_MANAGEMENT  The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.  O.SOUND_IMPLEMENTATION  The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.  O.THOROUGH_FUNCTIONAL_ TESTING  The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.  O.VULNERABILITY_ANALYSIS_ TEST  The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. | O.CHANGE_MANAGEMENT (ALC_CMC.4, ALC_CMS.4, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1, ACM_AUT.1) This objective plays a role in mitigating this threat in the same way that the flawed design threat is mitigated. By controlling who has access to the TOE's implementation representation and ensuring that changes to the implementation are analyzed and made in a controlled manner, the threat of intentional or unintentional errors being introduced into the implementation are reduced.  In addition to documenting the design so that implementers have a thorough understanding of the design, O.SOUND_IMPLEMENTATION (ADV_TDS.4, ADV_INT.1, (ADV_IMP.2, ALC_TAT.1) requires that the developer's tools and techniques for implementing the design are documented. Having accurate and complete documentation, and having the appropriate tools and procedures in the development process helps reduce the likelihood of unintentional errors being introduced into the implementation.  Although the previous three objectives help minimize the introduction of errors into the implementation, O.THOROUGH_FUNCTIONAL_TESTING (ATE_COV.2, ATE_FUN.1, ATE_DPT.3, ATE_IND.2) increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.  O.VULNERABILITY_ANALYSIS_TEST (AVA_VAN.4) helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation, and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing. Having an independent party perform a vulnerability analysis and conduct testing outside the scope of functional testing increases the likelihood of finding errors. |
| T.POOR_TEST  Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect | O.CORRECT_ TSF_OPERATION  The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. | Design analysis determines that TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security |

101

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| TOE behavior being undiscovered thereby causing potential security vulnerabilities. | O.THOROUGH_FUNCTIONAL_ TESTING<br><br>The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.<br><br>O.VULNERABILITY_ANALYSIS_ TEST<br><br>The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. | mechanisms must be performed during the evaluation of the TOE. O.THOROUGH_FUNCTIONAL_ TESTING (ATE_FUN.1, ATE_COV.2, ATE_DPT.3, ATE_IND.2) ensures that adequate functional testing is performed to demonstrate the TSF satisfies the security functional requirements and that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies. O.VULNERABILITY_ANALYSIS_TEST (AVA_VAN.4) addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing.<br><br>While these testing activities are necessary for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.CORRECT_ TSF_OPERATION (FPT_TST_(EXT).1, FPT_TST.1(1), FPT_TST.1(2)) ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software, including the cryptographic functions) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced. |
| T.REPLAY<br><br>A user may gain inappropriate access to the TOE by replaying authentication information. | O.REPLAY_DETECTION<br><br>The TOE will provide a means to detect and reject the replay of authentication data. | O.REPLAY_DETECTION (FPT_RPL.1) prevents a user from replaying authentication data. Prevention of replay of authentication data will counter the threat that a user will be able to record an authentication session between a trusted entity (administrative user or trusted IT entity) and then replay it to gain access to the TOE, as well as counter the ability of a user to act as another user. |
| T.RESIDUAL_DATA<br><br>A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. | O.RESIDUAL_INFORMATION<br><br>The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.<br><br>O.CRYPTOGRAPHIC_FUNCTIONS<br><br>The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. | O.RESIDUAL_INFORMATION (FDP_RIP.2) counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets sent in response to a request will not have residual data from another packet (potentially from another user) due to the padding of a packet.<br><br>O.CRYPTOGRAPHIC_FUNCTIONS (FCS_CKM.4 and FCS_CKM_(EXT).2) mitigates this threat by ensuring that the cryptographic data does not reside in a resource that has been used by the cryptographic functions and then reallocated to another process |
| T.RESOURCE_EXHAUSTION<br><br>A malicious process or user may block others from system resources (e.g., CPU time) via a resource exhaustion denial of service attack. | O.RESOURCE_SHARING<br><br>The TOE shall provide mechanisms that mitigate attempts to exhaust CPU time and available network connections provided by the TOE. | O.RESOURCE_SHARING (FRU_RSA.1(1)-(2), FMT_MTD.2(1)-(2)) mitigates this threat by requiring the TOE to provide controls relating to two different resources: CPU time and available network connections. The administrator is allowed to specify a percentage of processor time that is allowed to be used so that an attempt to exhaust the resource will |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | fail when it reaches the quota. This objective also addresses the denial-of-service attack of a user attempting to exhaust the connection-oriented resources by generating a large number of half-open connections (e.g., SYN attack). |
| T.SPOOFING<br><br>An entity may misrepresent itself as the TOE to obtain authentication data. | O.TRUSTED_PATH<br><br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with a authorized IT entity and not some other entity pretending to be a authorized IT entity. | It is possible for an entity other than the TOE (a subject on the TOE, or another IT entity on the network between the TOE and the end user) to provide an environment that may lead a user to mistakenly believe they are interacting with the TOE, thereby fooling the user into divulging identification and authentication information. O.TRUSTED_PATH (FTP_ITC_(EXT).1(1), FTP_ITC_(EXT).1(2), FTP_TRP_(EXT).1(1), FTP_TRP_(EXT).1(2)) mitigates this threat by ensuring users have the capability to ensure they are communicating with the TOE when providing identification and authentication data to the TOE. |
| T.MALICIOUS_TSF_ COMPROMISE<br><br>A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted). | O.DISPLAY_BANNER<br><br>The TOE will display an advisory warning regarding use of the TOE.<br><br>O.MANAGE<br><br>The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.<br><br>O.RESIDUAL_INFORMATION<br><br>The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.<br><br>O.SELF_PROTECTION<br><br>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.<br><br>O.TRUSTED_PATH<br><br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. | O.TRUSTED_PATH (FTP_TRP_(EXT).1(1), FTP_TRP_(EXT).1(2), FTP_ITC_(EXT).1(1), FTP_ITC_(EXT).1(2)) plays a role in addressing this threat by ensuring that there is a trusted communication path between the TSF and various users (remote administrators, relying parties (for authentication) and trusted IT entities (for performing replication, for instance)). This ensures the transmitted data cannot be compromised or disclosed during the duration of the trusted path. The protection offered by this objective is limited to TSF data, including authentication data and all data sent or received by trusted IT entities (a relying party's user data is not protected; only the authentication portion of the session is protected).<br><br>O.MANAGE (FMT_MTD.1(1)-(4), FMT_MSA.1, FMT_MOF.1(1)-(2), FMT_MTD.2(1)-(2)) provides the capability to restrict access to TSF to those that are authorized to use the functions. Satisfaction of this objective (and its associated requirements) prevents unauthorized access to TSF functions and data through the administrative mechanisms.<br><br>O.RESIDUAL_INFORMATION (FDP_RIP.2) is necessary to mitigate this threat by ensuring no TSF data remain in resources allocated to a user. Even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.<br><br>O.SELF_PROTECTION ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.<br><br>O.DISPLAY_BANNER (FTA_TAB.1) helps mitigate this threat by providing the Platform Administrator the ability to remove product information (e.g., product name, version number) from a banner that is displayed to users. Having product information about the TOE provides an |

103

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | attacker with information that may increase their ability to compromise the TOE. |
| **T.UNATTENDED_SESSION**<br><br>A user may gain unauthorized access to an unattended session. | **O.ROBUST_TOE_ACCESS**<br><br>The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate | O.ROBUST_TOE_ACCESS (FTA_SSL.1, FTA_SSL.2, FTA_SSL.3(1)-(2)) helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions are locked and remote sessions are dropped after a Platform Administrator-defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session. |
| **T.UNAUTHORIZED_ACCESS**<br><br>A user may gain access to user data for which they are not authorized according to the TOE security policy. | **O.MEDIATE**<br><br>The TOE must protect user data in accordance with its security policy.<br><br>**O.SELF_PROTECTION**<br><br>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. | O.MEDIATE (FDP_ACC.2, FDP_ACF.1) works to mitigate this threat by requiring that objects in the directory are protected using access control items. An access control item contains information about who is allowed to access an object, as well as the allowed modes of access. The settings present in the access control item selected in the access control decision process determine whether or not a user is authorized to access the object. It should be noted that multiple security policies can be (but do not *have* to be) in place in a single TOE, meaning that the process by which the target ACI is selected can be different for two different objects. It is required, however, that all objects be covered by this policy. Note that O.SELF_PROTECTION ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs**.** Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.<br><br>Because of the A.NO_GENERAL_PURPOSE assumption and the other requirements on the TOE, there is no requirement for a platform-level general-purpose access control policy. The only users that are required to have access to the platform are administrative users, and the policies that dictate their access are specified in other requirements (e.g., the FMT class). |
| **T.UNIDENTIFIED_ACTIONS**<br><br>The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach. | **O.AUDIT_REVIEW**<br><br>The TOE will provide the capability to selectively view audit information, and alert an administrator of identified potential security violations. | O.AUDIT_REVIEW (FAU_SAA.1-NIAP-0407, FAU_ARP.1, FAU_ARP_ACK_DIR_(EXT).1, FAU_ SAR.1(1)-(2), FAU_SAR.3) helps to mitigate this threat by providing a variety of mechanisms for monitoring the use of the system. The two basic ways audit review is performed is through analysis of the audit trail produced by the audit mechanism, and through the use of an automated analysis and alarm system.<br><br>For analyzing the audit trail, the TOE requires an Auditor role. This role is restricted to Audit record review and the deletion of the audit trail for maintenance purposes. A search and sort capability provides an efficient mechanism for the Audit Administrator to view pertinent audit information. In |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | addition to the local Auditor role, the TOE also has the capability to export the audit information to an external audit analysis tool (such as an intrusion detection system) for more detailed or composite audit analysis. |
| | | The TOE's audit analysis mechanism must consist of a minimum set of configurable audit events that could indicate a potential security violation. Thresholds for these events must be configurable by an appropriate administrative role. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures, set number directory access failures, self-test failures, etc.) and immediately notifies an administrator once an event has occurred or a set threshold has been met. If a potential security violation has been detected, the TOE displays a message that identifies the potential security violation to all administrative consoles. The consoles include the local TOE console and any active remote directory administrator sessions. If an administrator is not currently logged into the TOE, the message is stored and immediately displayed the next time an administrator logs into the TOE. This message is displayed and will remain on the screen until an administrator acknowledges the message. At this point, all administrators that have received the message will receive notification that the alarm has been acknowledged, who acknowledged the alarm, and the time that it was acknowledged. |
| | | In addition to displaying the potential security violation, the message must contain all audit records that generated the potential security violation. By enforcing the message content and display, this objective provides assurance that a TOE administrator will be notified of a potential security violation. |
| T.UNKNOWN_STATE<br><br>When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown. | O.MAINT_MODE<br><br>The TOE shall provide a mode from which recovery or initial startup procedures can be performed.<br><br>O.CORRECT_TSF_OPERATION<br><br>The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.<br><br>O.SOUND_DESIGN<br><br>The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented.<br><br>O.ROBUST_ADMIN_GUIDANCE<br><br>The TOE will provide administrators with the necessary information for secure delivery and management. | O.SOUND_DESIGN (ADV_FSP.5, ADV_TDS.4) works to mitigate this threat by requiring that the TOE developers provide accurate and complete design documentation of the security mechanisms in the TOE. By providing this documentation, the possible secure states of the TOE are described, thus enabling the administrator to return the TOE to one of these states during the recovery process.<br><br>O.MAINT_MODE (FPT_RCV.2) helps to mitigate this threat by ensuring that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. After a failure, the TOE enters a state that disallows operations and requires an administrator to follow documented procedures to return the TOE to a secure state.<br><br>O.CORRECT_TSF_OPERATION (FPT_TST_(EXT).1, FPT_TST.1(1) and FPT_TST.1(2)) counters this threat by ensuring that the TSF runs a suite of tests to successfully demonstrate the correct operation of the TSF (hardware and software) and the TSF's cryptographic components at initial startup of the TOE. In addition to ensuring that the TOE's security state can be verified, an administrator can verify the integrity of the TSF's data and stored code as well as the TSF's cryptographic data and stored code using the TOE- |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | provided cryptographic mechanisms.<br><br>O.ROBUST_ADMIN_GUIDANCE (ADV_OPE.1, AGD_PRE.1 1) provides administrative guidance for the secure start-up of the TOE as well as guidance to configure and administer the TOE securely. This guidance provides administrators with the information necessary to ensure that the TOE is started and initialized in a secure manor. The guidance also provides information about the corrective measure necessary when a failure occurs (i.e., how to bring the TOE back into a secure state). |
| P.ACCESS_BANNER<br><br>The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. | O.DISPLAY_BANNER<br><br>The TOE will display an advisory warning regarding use of the TOE. | O.DISPLAY_BANNER (FTA_TAB.1) satisfies this policy by ensuring that the TOE displays a Platform Administrator-configurable banner that provides all users with a warning about the unauthorized use of the TOE. This is required to be displayed before an interactive administrative session, since it does not make sense to display a banner for sessions involving directory requests from users, and those types of sessions are largely automated. |
| P.ACCOUNTABILITY<br><br>The authorized users of the TOE shall be held accountable for their actions within the TOE. | O.AUDIT_GENERATION<br><br>The TOE will provide the capability to detect and create records of security-relevant events associated with users.<br><br>O.AUDIT_PROTECTION<br><br>The TOE will provide the capability to protect audit information.<br><br>O.TIME_STAMPS<br><br>The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.<br><br>O.ROBUST_TOE_ACCESS<br><br>The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate | O.AUDIT_GENERATION (FAU_GEN.1-NIAP-0347, FAU_GEN.2-NIAP-410, FIA_USB.1, FAU_SEL.1-NIAP-0407) addresses this policy by providing an audit mechanism to record the actions of a specific user, as well as the capability for an administrator to "pre-select" audit events based on the user ID. The audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). Attributes used in the audit record generation process are also required to be bound to the subject, ensuring users are held accountable.<br><br>O.AUDIT_PROTECTION (FAU.SAR.2, FAU_STG.1-NIAP-0429, FAU_STG.3, FAU_STG.NIAP-0414-1-NIAO-0429-1, FMT_SMF.1) address this policy by providing an archive of the audit data so an administrator can look at a complete history of audit data.<br><br>O.TIME_STAMPS (FPT_STM.1, FMT_MTD.1(3)) plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (configured locally by the Platform Administrator or via a trusted IT entity, such as an NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID will also include the date and time that the event occurred.<br><br>O.ROBUST_TOE_ACCESS (FIA_UID.2, FIA_UAU.2, FIA_UAU.5) supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users. Note that although the TSF allows access by anonymous users (FIA_UAU.1), this objective (and hence the policy) does not apply to such users because they are not authenticated. |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| P.ADMIN_ACCESS<br><br>Administrators shall be able to administer the TOE both locally and remotely through protected communications channels. | O.ADMIN_ROLE<br><br>The TOE will provide administrator roles to isolate administrative actions.<br><br>O.TRUSTED_PATH<br><br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. | O.ADMIN_ROLE (FMT_SMR.2(1)-(2)) supports this policy by requiring the TOE to provide mechanisms (e.g., local authentication, remote authentication, means to configure and manage the TOE both remotely and locally) that allow remote and local administration of the TOE. This is not to say that everything that can be done by a local administrator must also be provided to the remote administrator. In fact, it may be desirable to have some functionality restricted to the local administrator.<br><br>O.TRUSTED_PATH (FTP_TRP_(EXT).1(1), FTP_TRP_(EXT).1(2), FTP_ITC_(EXT).1(1), FTP_ITC_(EXT).1(2)) satisfies this policy by requiring that each remote administrative and management session for all trusted users is authenticated and conducted via a secure channel. Additionally, all trusted IT entities (e.g., trusted peer directories, intrusion detection systems) connect through a protected channel, thus avoiding disclosure and spoofing problems. This objective works in conjunction with the IT environment objective, OE.TRUSTED_PATH, each providing one end of the trusted channel. |
| P.CRYPTOGRAPHY_VALIDATED<br><br>Where the TOE requires FIPS-approved security functions, only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key distribution, and random number generation services). | O.CRYPTOGRAPHY_VALIDATED<br><br>The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions. | O.CRYPTOGRAPHY_VALIDATED FCS_BCM_(EXT).1 FCS_CKM.1(1) FCS_CKM.1(2) FCS_CKM.2 FCS_CKM.4 FCS_CKM_(EXT).2 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_COP_(EXT).1 implements this policy by requiring the TOE to implement NIST FIPS-validated cryptographic services. The objective requires that the functions needed by the TOE are FIPS approved, and further that they are available in a FIPS-approved mode of operation of the cryptomodule. |
| P.CRYPTOGRAPHIC_FUNCTIONS<br><br>The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. | O.CRYPTOGRAPHIC_FUNCTIONS<br><br>The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. | O.CRYPTOGRAPHIC_FUNCTIONS FCS_BCM_(EXT).1 FCS_CKM.1(1) FCS_CKM.1(2) FCS_CKM.2 FCS_CKM.4 FCS_CKM_(EXT).2 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_COP_(EXT).1 implements this policy, requiring a combination of FIPS-validation and non-FIPS-validated cryptographic mechanisms that are used to provide encryption/decryption services, as well as digital signature functions. Functions include symmetric encryption and decryption, digital signatures, as well as key generation and establishment functions. |
| P.DISTRIBUTED_DIRECTORY_SUPPORT<br><br>Directories shall be able to support replication. To support replication directories shall be able to replicate (both produce and consume) definable subtrees to other directories (peer trusted directories). Directories shall be able to authenticate using a distributed authentication mechanism. | O.DISTRIBUTED_DIRECTORY_SUPPORT<br><br>The TSF shall be able to replicate definable subtrees to (produce) and accept replications of definable subtrees from (consume) other directories. The TSF shall be to authenticate using a distributed authentication mechanism. | O.DISTRIBUTED_DIRECTORY_SUPPORT (FDD_RPL_(EXT).1, FIA_UAU.5, FTP_ITC_(EXT).1(1) FTP_ITC_(EXT).1(2) ) implements the policy by providing the replication service. This service allows replication of subtrees, as well as the ability for the TSF to either produce or consume the replicated data. Security attributes are associated with the replicated data to ensure a consistent enforcement of the security policy.<br><br>The policy is also implemented by the TSF distributed authentication mechanism. In addition, the TOE can be trusted to be the introducer or presenter to a peer directory by ensuring the integrity |

| Threat/Policy | Objectives Addressing the Threat | Rationale |
|---|---|---|
| | | and confidentiality of the user authentication data. |
| P.NONREPUDIATION<br><br>The TOE must provide non-repudiation services for transmitted and received repository data. The non-repudiation services include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that evidence of receipt the TOE is waiting for is overdue. | O.NONREPUDIATION<br><br>At the option of an administrator, the TSF must be able to provide non-repudiation services for transmitted and received repository data. These services must include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that the evidence of receipt the TOE is waiting for is overdue. | O.NONREPUDIATION (FCO_PRA_(EXT).1) supplies the non-repudiation mechanism on both origin (when the TSF is acting as the producer) and receipt (when the TSF is acting as the consumer) of the replicated directory information. The services provided include the ability to both generate and display the evidence used to provide the originator of the data as well as the fact that the data were received, and functionality to notify the Security Administrator when a timely response is not received. This objective works in conjunction with the IT environment objective, OE.NONREPUDIATION, where the IT environment provides the evidence of receipt when the TOE is the originator. |
| P.VULNERABILITY_ ANALYSIS_TEST<br><br>The TOE must undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential. | O.VULNERABILITY_ANALYSIS_ TEST<br><br>The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. | O.VULNERABILITY_ANALYSIS_TEST (AVA_VAN.4) satisfies this policy by ensuring that an independent analysis is performed on the TOE and penetration testing based on that analysis is performed. Having an independent party perform the analysis helps ensure objectivity and eliminates preconceived notions of the TOE's design and implementation that may otherwise affect the thoroughness of the analysis. The level of analysis and testing requires that an attacker with a moderate attack potential cannot compromise the TOE's ability to enforce its security policies. |

## 6.2 RATIONALE FOR THE SECURITY OBJECTIVES AND SECURITY FUNCTIONAL REQUIREMENTS FOR THE ENVIRONMENT

All but two of the security objectives for the environment, OE.TRUSTED_PATH and OE.EVIDENCE_OF_RECEIPT_OF_REPLICA_DATA, are restatements of an assumption found in Section 3. Because of this, those security objectives for the environment completely capture the assumptions, and are therefore suitable for covering the assumptions listed in the PP.

The IT security objective OE.TRUSTED_PATH(FTP_TRP_(EXT).1(3), FTP_TRP_(EXT).1(4), FTP_ITC_(EXT).1(3), FTP_ITC_(EXT).1(4)) is necessary to satisfy the policy P.ADMIN_ACCESS. This IT security objective for the environment works in conjunction with the TOE security objective O.TRUSTED_PATH, each providing one end of a trusted channel, to ensure there is a trusted communications channel for remote administrative and management sessions for all trusted users and authorized IT entities (e.g., trusted peer directories, intrusion detection system), thus avoiding disclosure and spoofing problems. OE.TRUSTED_PATH maps to the IT environmental iterated requirements FTP_TRP_(EXT).1(3) FTP_TRP_(EXT).1(4), ensuring that an administrator and replying parties authenticating with a password can be assured that they are communicating with the TOE. It also maps to FTP_ITC_(EXT).1(3), and FTP_ITC_(EXT).1(4) ensuring that encryption is used to create the trusted communication channel between trusted external IT entities and the TOE.

The IT security objective OE.EVIDENCE_OF_RECEIPT_OF_REPLICA_DATA (FCO_PRA_(EXT).1(2)) is necessary to satisfy the policy P.NONREPUDIATION. This IT security objective for the environment works in conjunction with the TOE security objective O.NONREPUDIATION, where OE.EVIDENCE_OF_RECEIPT_OF_REPLICA_DATA provides the evidence of receipt when the TOE is the originator, thus providing non-repudiation of transmitted repository data. OE.EVIDENCE_OF_RECEIPT_OF_REPLICA_DATA maps to the IT environmental iterated requirement FCO_PRA_(EXT).1(2) ensuring the IT entity in the environment provides the evidence of receipt.

## 6.3 RATIONALE FOR TOE SECURITY REQUIREMENTS

### Table 6.2 – Rationale for TOE Security Requirements

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| O.ROBUST_ADMIN_GUIDANCE<br><br>The TOE will provide administrators with the necessary information for secure delivery and management. | ALC_DEL.1<br><br>AGD_PRE.1<br><br>AGD_OPE.1 | ALC_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a *clean* (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE.<br><br>The AGD_PRE.1 requirement ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.<br><br>The AGD_OPE.1 requirement mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's ruleset and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. AGD_OPE.1 is also intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). |
| O.ADMIN_ROLE<br><br>The TOE will provide administrator roles to isolate administrative actions. | FMT_SMR.2(1)<br><br>FMT_SMR.2(2) | FMT_SMR.2 requires that four roles exist for administrative actions: the Security Administrator, who is responsible for configuring most security-relevant parameters on the TOE; the Cryptographic Administrator, who is responsible for managing the |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | security data that is critical to the cryptographic operations; the Auditor, who is responsible for reading and deleting the audit trail; and one or more directory managers, who is able to perform directory operations on some portion of the directory hierarchy. The security administrator defines a directory manager's scope of control. The TSF is able to associate a human user with one or more roles and these roles isolate administrative functions in that the functions of these roles do not overlap (except for the directory manager roles, discussed below). It is true that the design of some systems could enable a rogue security administrator to manipulate cryptographic data by, for instance, writing directly to kernel memory. While this scenario is a security concern, this objective does not counter that aspect of T.ADMIN_ROGUE. If a security administrator were to perform such an action, the auditing requirements (along with the audit trail protection requirements) afford some measure of detectability of the rogue platform administrator's actions. <br><br> The manager roles, unlike the roles in FMT_SMR.2(1), are not required to have totally isolated functions. Instead, each directory manager will have a subset of the functionality, as well as a subset of the scope of control, of the security administrator. Thus, if the directory manager is the rogue admin, the damage will be isolated to the portion of the directory hierarchy over which the directory manager has control, and will likely not affect the rest of the directory. The security administrator, as mentioned above, is responsible for defining the scope of control for the directory managers. |
| **O.AUDIT_GENERATION** <br><br> The TOE will provide the capability to detect and create records of security-relevant events associated with users. | FAU_GEN.1-NIAP-0347 <br><br> FAU_GEN.2-NIAP-0410 <br><br> FIA_USB.1 <br><br> FAU_SEL.1-NIAP-0407 | FAU_GEN.1-NIAP-0347 defines the set of events that the TOE must be capable of recording. This requirement ensures that an administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. <br><br> FAU_GEN.2-NIAP-410 ensures that the audit records associate a user identity with the auditable event. Although the FIA_ATD.1(*) requirements mandate that a "userid" be used to represent a user identity, the TOE developer is able to associate different types of userids with different users in order to meet this objective. <br><br> FAU_SEL.1-NIAP-0407 allows the selected administrator(s) to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism and providing the ability to focus on the actions of an individual user. In addition, the requirement has been refined to require that the audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions. <br><br> FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | trail may not always have the proper identity of the subject that causes an audit record to be generated (anonymous relying parties). |
| O.AUDIT_PROTECTION<br><br>The TOE will provide the capability to protect audit information. | FAU_STG.1-NIAP-0429<br><br>FAU_SAR.2<br><br>FAU_STG.NIAP-0414-1-NIAP-0429<br><br>FAU_STG.3<br><br>FMT_SMF.1 | FAU_SAR.2 restricts the ability to read the audit trail to the Auditor, thus preventing the disclosure of the audit data to any other user. However, the TOE is not expected to prevent the disclosure of audit data if it has been archived or saved in another form (e.g., moved or copied to an ordinary file).<br><br>The FAU_STG family dictates how the audit trail is protected. FAU_STG.1-NIAP-0429 restricts the ability to delete audit records to the Auditor; or if the option of overwriting old audit records is chosen by the Platform/Directory Administrator in FAU_STG.NIAP-0414-1-NIAP-0429, the audit data may be deleted/overwritten. Since the auditor is trusted to review the audit data, the threat being countered is that the platform/directory administrator does something malicious and then attempts to conceal it by configuring the audit log to overwrite old records. Presumably the platform/directory administrator would then attempt to fill up the audit log in order to overwrite the thing they just did, as well as the fact that the they reconfigured the audit log overwrite action. The auditor would hopefully notice this activity and detect the fact that the platform/directory administrator was performing illicit activities. The fact that the platform/directory administrator does not directly have the ability to delete the audit records helps ensure that audit records are kept until the Auditor deems they are no longer necessary. FAU_STG.1-NIAP-0429 also ensures that no one has the ability to modify audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the audit trail is maintained.<br><br>FAU_STG.3 requires that the administrators be alerted when the audit trail exceeds a capacity threshold established by the Security Administrator. In addition, an audit record is cut which will trigger the analysis performed in FAU_SAA, resulting in an FAU_ARP alarm being issued. This ensures that an administrator has the opportunity to manage the audit trail before it becomes full and the avoiding the possible loss of audit data.<br><br>FAU_STG.NIAP-0414-1-NIAP-0429 allows the Security Administrator to configure the TOE so that if the audit trail does become full, either the TOE will prevent any events from occurring (other than actions taken by the administrator) that would generate an audit record or the audit mechanism will overwrite the oldest audit records with new records.<br><br>FMT_SMF.1 requires the TOE to provide an administrator with a facility to backup, recover and archive audit data ensuring the ability to recover corrupted audit records, and access to a complete history of audit information. |
| O.AUDIT_REVIEW<br><br>The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations. | FAU_SAA.1-NIAP-0407<br><br>FAU_ARP.1<br><br>FAU_ARP_ACK_DIR_(EXT).1<br><br>FAU_SAR.3<br><br>FAU_SAR.1(1)<br><br>FAU_SAR.1(2) | FAU_SAA.1-NIAP-0407 defines the events (or rules) that indicate a potential security violation and will generate an alarm. The triggers for these events are largely configurable by the Security Administrator. Some rules are not configurable, or configurable by the cryptographic administrator.<br><br>FAU_ARP.1 requires that the alarm be displayed at the local administrative console and at the remote administrative console(s) when auditor and security administrative session(s) exists. For alarms at remote consoles, the alarm is sent either during an established session or upon session establishment (as long as the alarm has not been acknowledged). This is required to increase the likelihood that the alarm will be received as soon as possible. This |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | requirement also dictates the information that must be displayed with the alarm. The potential security violation is identified in the alarm, as are the contents of the audit records of the events that accumulated and triggered the alarm. The information in the audit records is necessary it allows the administrators to react to the potential security violation without having to search through the audit trail looking for the related events. |
| | | FAU_ARP_ACK_(EXT).1 requires that an alarm generated by the mechanism that implements the FAU_ARP requirement be maintained until an administrator acknowledges it. This ensures that the alarm message will not be obstructed and the administrators will be alerted of a potential security violation. Additionally, this requires that the acknowledgement be transmitted to users that received the alarm, thus ensuring that that set of administrators knows that the user specified in the acknowledgement message has addressed the alarm. |
| | | FAU_SAR.1 (both iterations) is used to provide both the auditor and an external audit analysis function the capability to read all the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the end user (auditor or external system) to interpret the audit trail. It is expected that the audit information be presented in such a way that the end user can examine an audit record and have the appropriate information (that required by FAU_GEN.2-NIAP-410) presented together to facilitate the analysis of the audit review. Ensuring the audit data are presented in an interpretable format will enhance the ability of the entity performing the analysis to identify potential security violations. |
| | | FAU_SAR.3 complements FAU_SAR.1 by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrators be able to establish the audit review criteria based on a userid and role so that the actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates and times provides the capability to facilitate the administrator's review of incidents that may have taken place at a certain time. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria. |
| O.CHANGE_MANAGEMENT<br><br>The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development. | ALC_CMC.4<br><br>ALC_CMS.4<br><br><br>ALC_DVS.1<br><br>ALC_FLR.2<br><br>ALC_LCD.1 | ALC_CMC.4 contributes to this objective by requiring the developer have a configuration management plan that describes how changes to the TOE and its evaluation deliverables are managed. The developer is also required to employ a configuration management system that operates in accordance with the CM plan and provides the capability to control who on the development staff can make changes to the TOE and its developed evidence. This requirement also ensures that authorized changes to the TOE have been analyzed and the developer's acceptance plan describes how this analysis is performed and how decisions to incorporate the changes to the TOE are made. ALC_CMC.4 also requires that the CM system use an automated means to control changes made to the TOE. If automated tools are used by the developer to analyze or track changes made to the TOE, those automated tools must be described.<br><br>ALC_CMS.4 is necessary to define what items must be under the control of the CM system. This requirement ensures that the TOE implementation representation, design documentation, test documentation (including the executable test suite), user and administrator guidance, CM documentation and security flaws are |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | tracked by the CM system. |
| | | ALC_DVS.1 requires the developer describe the security measures they employ to ensure the integrity and confidentiality of the TOE are maintained. The physical, procedural, and personnel security measures the developer uses provides an added level of control over who and how changes are made to the TOE and its associated evidence. |
| | | ALC_FLR.2 plays a role in satisfying the "analyzed" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws. |
| | | ALC_LCD.1 requires the developer to document the life-cycle model used in the development and maintenance of the TOE. This life-cycle model describes the procedural aspects regarding the development of the TOE, such as design methods, code or documentation reviews, how changes to the TOE are reviewed and accepted or rejected. |
| O.CORRECT_ TSF_OPERATION<br><br>The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. | FPT_TST_(EXT).1<br><br>FPT_TST.1(1)<br><br>FPT_TST.1(2) | O_CORRECT_TSF_OPERATION requires two extended functional requirements: FPT_TST_(EXT).1 for portions of the TOE that are not related to cryptographic functionality, FPT_TST.1(1) and FPT_TST.1(2) for those that are. These functional requirements provide the end user with the capability to ensure the TOE's security mechanisms continue to operate correctly in the field. |
| | | From the perspective of non-cryptographic hardware and software, FPT_TST_(EXT).1 provides the necessary functionality. The first element ensures end user tests exist to demonstrate the correct operation of the security mechanisms required by the TOE that is provided by the hardware. Hardware failures could render a TOE's software ineffective in enforcing its security policies and this requirement provides the end user the ability to discover any failures in the hardware security mechanisms. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data; if TSF data are corrupt the TOE may not correctly enforce its security policies. Some TSF data, however, is always changing (for instance, a file containing audit records) and therefore is not suitable for integrity checking mechanisms. These data are identified so that the administrator can understand the limitations of the mechanism. In order to protect the TSF code and data, the second and third elements require the use of a cryptographic mechanism to ensure that the TSF data, as well as the executable TSF code, have not been corrupted. |
| | | FPT_TST.1(1) and FPT_TST.1(2) addresses the critical nature and specific handling of the cryptographic-related TSF mechanisms. The cryptomodules have self-tests that are validated as part of the FIPS 140-2 process; this requirement ensures that those tests are invoked commensurate with the requirements on self-tests for other parts of the TOE. Additionally, because key material is critical to the security provided by cryptographic mechanisms, the TSF is required to provide a capability to run the self-tests after generation of a key to help ensure that an undetected failure did not compromise the integrity of the key that |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | was just generated. |
| O.CRYPTOGRAPHY_VALIDATED<br><br>The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions. | FCS_CKM.1(1)<br><br>FCS_CKM.1(2)<br><br>FCS_CKM.2<br><br>FCS_CKM.4<br><br>FCS_CKM_(EXT).2<br><br>FCS_BCM_(EXT).1<br><br>FCS_COP.1(1)<br><br>FCS_COP.1(2)<br><br>FCS_COP.1(3)<br><br>FCS_COP.1(4)<br><br>FCS_COP_(EXT).1 | Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules implemented in hardware, in software, or in hardware/software combinations [FCS_BCM_(EXT).1]. The cryptographic services offered by this baseline capability are augmented and customized in the TOE to support medium robustness environments. These TOE services are based primarily upon functional security requirements in the areas of key management and cryptographic operations. In the area of key management there are functional requirements that address the generation of symmetric keys [FCS_CKM.1 (1)], and the generation of asymmetric keys [FCS_CKM.1 (2)]; methods of manual and automated cryptographic key distribution [FCS_CKM.2]; cryptographic key destruction [FCS_CKM.4]; techniques for cryptographic key validation and packaging [FCS_CKM.1]; and cryptographic key handling and storage [FCS_CKM_(EXT).2]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; cryptographic key agreement [FCS_COP.1 (4)]; and improved random number generation [FCS_COP_(EXT).1]. |
| O.CRYPTOGRAPHIC_FUNCTIONS<br><br>The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations. | FCS_CKM.1(1)<br><br>FCS_CKM.1(2)<br><br>FCS_CKM.2<br><br>FCS_CKM.4<br><br>FCS_CKM_(EXT).2<br><br>FCS_BCM_(EXT).1<br><br>FCS_COP.1(1)<br><br>FCS_COP.1(2)<br><br>FCS_COP.1(3)<br><br>FCS_COP.1(4)<br><br>FCS_COP_(EXT).1 | Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules implemented in hardware, in software, or in hardware/software combinations [FCS_BCM_(EXT).1]. The cryptographic services offered by this baseline capability are augmented and customized in the TOE to support medium robustness environments. These TOE services are based primarily upon functional security requirements in the areas of key management and cryptographic operations. In the area of key management there are functional requirements that address the generation of symmetric keys [FCS_CKM.1 (1)], and the generation of asymmetric keys [FCS_CKM.1 (2)]; methods of manual and automated cryptographic key distribution [FCS_CKM.2]; cryptographic key destruction [FCS_CKM.4]; techniques for cryptographic key validation and packaging [FCS_CKM.1]; and cryptographic key handling and storage [FCS_CKM_(EXT).2]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; cryptographic key agreement [FCS_COP.1 (4)]; and improved random number generation [FCS_COP_(EXT).1]. |
| O.DISPLAY_BANNER<br><br>The TOE will display an advisory warning regarding use of the TOE. | FTA_TAB.1 | FTA_TAB.1 meets this objective by requiring the TOE display a Platform Administrator-defined banner before an administrator can establish an interactive session. This banner is under complete control of the Platform Administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire. |
| O.DOCUMENT_KEY_ LEAKAGE<br><br>The bandwidth of channels that can be used to compromise key material shall be documented. | AVA_CCA_(EXT).2 | AVA_CCA_(EXT).2 requires that a covert channel analysis be performed on the entire TOE to determine the bandwidth of possible cryptographic key leakage. While there are no requirements to limit the bandwidth, the results of this analysis will provide useful guidance on what the specified lifetime of the cryptographic keys should be in order to reduce the damage due to |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | a key compromise. |
| O.THOROUGH_FUNCTIONAL_TESTING<br><br>The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements. | ATE_COV.2<br>ATE_FUN.1<br>ATE_DPT.3<br>ATE_IND.2 | In order to satisfy O.THOROUGH_FUNCTIONAL_ TESTING, the ATE class of requirements is necessary. The component ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage. In addition, the developer must provide the test suite executables and source code, which are used for independently verifying the test suite results and in support of the test coverage analysis activities. ATE_COV.2 requires the developer to provide a test coverage analysis that demonstrates the TSFI are completely addressed by the developer's test suite. While exhaustive testing of the TSFI is not required, this component ensures that the security functionality of each TSFI is addressed. This component also requires an independent confirmation of the completeness of the test suite, which aids in ensuring that correct security relevant functionality of a TSFI is demonstrated through the testing effort. ATE_DPT.3 requires the developer to provide a test coverage analysis that demonstrates depth of coverage of the test suite. This component complements ATE_COV.2 by ensuring that the developer takes into account the high-level and low-level design when developing their test suite. Since exhaustive testing of the TSFI is not required, ATE_DPT.3 ensures that subtleties in TSF behavior that are not readily apparent in the functional specification are addressed in the test suite. ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. This component also requires an independent party to attempt to craft functional tests that address functional behavior that is not demonstrated in the developer's test suite. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated. |
| O.MAINT_MODE<br><br>The TOE shall provide a mode from which recovery or initial startup procedures can be performed. | FPT_RCV.2 | This objective is met by using the FPT_RCV.2 requirement, which ensures that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. Upon the failure of the TSF self-tests the TOE will no longer be assured of enforcing its security policies. Therefore, the TOE enters a state that operations and requires an administrator to follow documented procedures that instruct them on to return the TOE to a secure state. These procedures may include running diagnostics of the hardware, or utilities that may correct any integrity problems found with the TSF data or code. Solely specifying that the administrator reload and install the TOE software from scratch, while might be required in some cases, does not meet the intent of this requirement. |
| O.MANAGE<br><br>The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. | FMT_MSA.1<br>FMT_MOF.1(1)<br>FMT_MOF.1(2)<br>FMT_MTD.1(1)<br>FMT_MTD.1(2)<br>FMT_MTD.1(3)<br>FMT_MTD.1(4)<br>FMT_MTD.2(1)<br>FMT_MTD.2(2) | The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirement's rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions.<br><br>FMT_MSA.1 provides the Security Administrator or Directory Manager the capability to manipulate the security attributes of the objects in their scope of control that determine the access policy for directory objects.<br><br>There are several functions in the TSF that need to be enabled or disabled: the ability to provide verification evidence for certain directory objects; the ability to replicate portions of the directory, either in a producer role or a consumer role; the ability to detect attempts to replay operations sent by a relying party; and the |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | FMT_SMF.1 | ability to enable the cryptographic module self-tests to be run after generation of a key. The use of these functions is specified and restricted by the FMT_MOF.1 iterations. |
| | | The requirement FMT_MTD.1(1) is intended to be used by the ST author, with possible iterations, to address TSF data that has not already been specified by other FMT requirements. This is necessary because the ST author may add TSF data in assignments that cannot be addressed ahead of time by the PP authors. This requirement specifies that the manipulation of these data be restricted to the security administrator. |
| | | FMT_MTD.1(2) provides the Cryptographic Administrator, and only the Cryptographic Administrator, the ability to modify the cryptographic security data. This allows the Cryptographic Administrator to change the critical data that affects the TOE's ability to perform its cryptographic functions properly. |
| | | FMT_MTD.1(3) provides the capability of setting the date and time that is used to generate time stamps to the Security Administrator or a trusted IT entity (authorized data manager). It is important to allow this functionality, due to clock drift and other circumstances, but the capability must be restricted. A trusted IT entity is allowed in the selection made by the ST author to take in account the use of an NTP server or some other service that provides time information without human intervention. |
| | | FMT_MTD.1(4) addresses the capabilities of data managers, who have responsibilities for security data management for sub-portions of the set of TSF data (for example, the platform clock time, sub-hierarchies of the directory). The scope of a data manager's responsibility is set by a security administrator, but they are expected to manage the entities in their scope of control without reliance on the security administrator. |
| | | FMT_MTD.2(1), FMT_MTD.2(2) restrict the setting of limits on the processor time and network connection resources, respectively, to an administrator. This capability allows an administrator to control the resources consumed by to provide a flexible policy with respect to denial of service attacks. |
| | | FMT_SMF.1 requires the TOE to provide a backup and restore capability for administrators to use to enable recovery of TSF data. |
| O.MEDIATE<br><br>The TOE must protect user data in accordance with its security policy. | FDP_ACC.2<br><br>FDP_ACF.1 | The FDP_ACC.2 and FDP_ACF.1 requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation of access to the directory takes place. Because of the A.NO_GENERAL_PURPOSE assumption the no access control policy (for relying parties) needs to be defined for platform resources. |
| | | FDP_ACC.2 specifies that the subjects under control of the policy are directory managers and relying parities, and that all operations that involve access to (minimally) the RI entries, RI attributes, and RI attribute values are controlled by the policy. These objects contain the user data to be protected. |
| | | FDP_ACF.1 details the manner in which the user data are to be protected. The basics called for by the requirement is to match a set of attributes associated with a subject to a set of "access control items" associated with the object they wish to access; all applicable ACIs need to grant access in order for the subject to perform the operation on the object. The details of how the ACIs are collected and the specific operations supported are specified in the ST, and with the attributes define the security policy to be enforced. Setting the attributes (implementing the security policy) |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | is a function of the directory administrator or directory manager. |
| O.REPLAY_DETECTION<br><br>The TOE will provide a means to detect and reject the replay of authentication data. | FPT_RPL.1 | The O.REPLAY_DETECTION objective is satisfied by FPT_RPL.1, which requires the TOE to detect and reject the attempted replay of authentication data from a remote user (administrator or relying party). This is sufficient to meet the objective because no untrusted users have local access to the TOE, thus there is no way to capture nor replay authentication data for a local session. |
| O.RESIDUAL_INFORMATION<br><br>The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. | FDP_RIP.2 | FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to build network packets containing replies to relying party requests is either cleared or that some buffer management scheme be employed to prevent the contents of a packet being disclosed in a subsequent packet (e.g., if padding is used in the construction of a packet, it must not contain another user's data or TSF data). |
| O.RESOURCE_SHARING<br><br>The TOE shall provide mechanisms that mitigate attempts to exhaust CPU time and available network connections provided by the TOE. | FRU_RSA.1(1)<br><br>FRU_RSA.1(2)<br><br>FMT_MTD.2(1)<br><br>FMT_MTD.2(2) | While an availability security policy does not explicitly exist, FRU_RSA.1 is used to mitigate potential resource exhaustion attempts. In order to mitigate the CPU exhaustion attempt, FRU_RSA.1(1) is included. This requires that the CPU time being consumed by a relying party must be limited to an amount specified by the security administrator (FMT_MTD.2(1)), and actions taken when an attempt is made are specified in FMT_MTD.2(1). This requirement takes into account all CPU resources being consumed by a user (relying party), and not just a single subject.<br><br>FRU_RSA.1(2) was used to reduce the impact of an attempt being made to exhaust transport-layer representation implementation artifacts (e.g., the TCP "half-open connection" attack).<br><br>This requirement indicates that a time period must exist when maximum quota (which is defined by the ST) is met or surpassed. Although this requirement (unlike the two previous requirements) does not mandate that the administrator be able to set this time period, FMT_MTD.2(2) restricts this functionality should the TOE implement it. FMT_MTD.2(2) also indicates (when filled in by the ST author) what action is to be taken when the quota is reached. |
| O.SELF_PROTECTION<br><br>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. | ADV_ARC.1 | ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation. |
| O.SOUND_DESIGN<br><br>The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented. | ADV_ARC.1<br><br>ADV_INT.1<br><br>ADV_FSP.5<br><br>ADV_TDS.4 | There are two different perspectives for this objective. One is from the developer's point of view and the other is from the evaluator's. The ADV class of requirements is levied to aide in the understanding of the design for both parties, which ultimately helps to ensure the design is sound.<br><br>ADV_ARC.1 The security architecture description will be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document (ADV_TDS.4). It will describe the security domains maintained by the TSF consistently with the SFRs as well as how the TSF initialization process is secure. The security architecture |

| Objective | Requirements Addressing the Objective | Rationale |
|-----------|----------------------------------------|-----------|
| | | description will demonstrate that the TSF protects itself from tampering and that the TSF prevents bypass of the SFR-enforcing functionality. |
| | | ADV_INT.1 ensures that the design of the TOE has been performed using good software engineering design principles that require a modular design of the TSF. Modular code increases the developer's understanding of the interactions within the TSF, which in turn, potentially reduces the amount of errors in the design. Having a modular design is imperative for evaluator's to gain an appropriate level of understanding of the TOE's design in a relatively short amount of time. The appropriate level of understanding is dictated by other assurance requirements in this PP (e.g., ATE_DPT.3, AVA_CCA_(EXT).2, AVA_VAN.4). |
| | | ADV_FSP.5 requires that the interfaces to the TSF be completely specified. In this TOE, a complete specification of the network interface (including the network interface card) is critical in understanding what functionality is presented to untrusted users and how that functionality fits into the enforcement of security policies. Some network protocols have inherent flaws and users have the ability to provide the TOE with network packets crafted to take advantage of these flaws. The routines/functions that process the fields in the network protocols allowed (e.g., TCP, UPD, ICMP, directory-specific protocols such as LDAP) must fully specified: the acceptable parameters, the errors that can be generated, and what, if any, exceptions exist in the processing. The functional specification of the hardware interface (e.g., network interface card) is also extremely critical. Any processing that is externally visible performed by NIC must be specified in the functional specification. Having a complete understanding of what is available at the TSF interface allows one to analyze this functionality in the context of design flaws. |
| | | TDS.4 - Provides a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.  The design will describe: the structure of the TOE in terms of subsystems; the TSF in terms of modules; identify all subsystems of the TSF; provide a description of each subsystem of the TSF; a description of the interactions among all subsystems of the TSF; a mapping from the subsystems of the TSF to the modules of the TSF; describe each SFR-enforcing module in terms of its purpose; describe each SFR-enforcing module in terms of its SFR-related interfaces; return values from those interfaces, and called interfaces to other modules; describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules; the mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.  The design, as required by ADV_TDS.4 , provides the evaluator with the details of the TOE's design and describes at a module level how the design of the TOE addresses the SFRs. This level of description provides the detail of how modules interact within the TOE and if a flaw exists in the TOE's design. This requirement also mandates that the interfaces presented by modules be specified. Having knowledge of the parameters a module accepts, the errors that can be returned and a description of how the module works to support the security policies allows the design to be understood at its lowest level. ADV_TDS.4 also o ensures that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design analysis to ensure design decisions are realized at all |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | levels of the design. |
| | | |
| O.SOUND_IMPLEMENTATION<br><br>The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented. | ADV_TDS.4<br><br>ADV_IMP.1<br><br>ADV_INT_ 1<br><br>ALC_TAT.1 | TDS.4 - Provides a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.  The design will describe: the structure of the TOE in terms of subsystems; the TSF in terms of modules; identify all subsystems of the TSF; provide a description of each subsystem of the TSF; a description of the interactions among all subsystems of the TSF; a mapping from the subsystems of the TSF to the modules of the TSF; describe each SFR-enforcing module in terms of its purpose; describe each SFR-enforcing module in terms of its SFR-related interfaces; return values from those interfaces, and called interfaces to other modules; describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules; the mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.  The design, as required by ADV_TDS.4 , provides the evaluator with the details of the TOE's design and describes at a module level how the design of the TOE addresses the SFRs. This level of description provides the detail of how modules interact within the TOE and if a flaw exists in the TOE's design. This requirement also mandates that the interfaces presented by modules be specified. Having knowledge of the parameters a module accepts, the errors that can be returned and a description of how the module works to support the security policies allows the design to be understood at its lowest level. ADV_TDS.4 also o ensures that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design analysis to ensure design decisions are realized at all levels of the design.<br><br>ADV_IMP.1, ensures evaluators have full access to the source code. If the evaluators are limited in their ability to analyze source code they may not be able to determine the accuracy of the implementation or the adequacy of the documentation. Often times it is difficult for an evaluator to identify the complete sample of code they wish to analyze. Often times looking at code in one subsystem may lead the evaluator to discover code they should look at in another subsystem. Rather than require the evaluator to "re-negotiate" another sample of code, the complete implementation representation is required.<br><br>When performing the activities associated with the ADV_INT.1 requirement, the evaluators will ensure that the architecture of the implementation is modular and consistent with the architecture presented in the low-level design. Having a modular implementation provides the evaluators with the ability to more easily assess the accuracy of the implementation, with respect to the design. If the implementation is overly complex (e.g., circular dependencies, not well understood coupling, reliance on side-effects) the evaluator may not have the ability to assess the accuracy of the implementation.<br><br>ALC_TAT.1 provides evaluators with information necessary to understand the implementation representation and what the |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | resulting implementation will consist of. Critical areas (e.g., the use of libraries, what definitions are used, compiler options) are documented so the evaluator can determine how the implementation representation is to be analyzed. |
| O.TIME_STAMPS<br><br>The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. | FPT_STM.1<br><br>FMT_MTD.1(3) | FPT_STM.1 requires that the TOE be able to provide reliable time stamps for its own use and therefore, partially satisfies this objective. Time stamps include date and time and are reliable in that they are always available to the TOE, and the clock must be monotonically increasing.<br><br>FMT_MTD.1(3) satisfies the rest of this objective by providing the capability to set the time used for generating time stamps to either the Security Administrator, trusted IT entity, or both. The authorized IT entity was included as an option for the possible use of an NTP server to set the TOE's time. |
| O.DISTRIBUTED_DIRECTORY_SUPPORT<br><br>The TSF shall be able to replicate definable subtrees to (produce) and accept replications of definable subtrees from (consume) other directories. The TSF shall be to authenticate using a distributed authentication mechanism. | FDD_RPL_(EXT).1<br><br>FIA_UAU.5<br><br>FTP_ITC.1(1&2) | FDD_RPL_(EXT).1 is the primary requirement concerning replication. This requirement specifies that the directory administrator controls the subtree and peer directory involved in the replication action. It also specifies that the security attributes be associated with the replicated information so that the security policy can be preserved. The requirement calls for the TOE to be able to act in both the producer role as well as the consumer role.<br><br>FIA_UAU.5 requires the TSF be able to authenticate a relying party using $3^{rd}$ party presentation or introduction from a peer trusted directory. When it's the introducer or presenter, the TSF provides a domain that protects itself from untrusted users, and requires a trusted channel for communication with a peer trusted directory to ensure the integrity and confidentiality of the user authentication data. |
| O.ROBUST_TOE_ACCESS<br><br>The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate | FTA_TSE.1<br><br>FIA_UID.2<br><br>FTA_SSL.1<br><br>FTA_SSL.2<br><br>FTA_SSL.3(1)<br><br>FTA_SSL.3(2)<br><br>AVA_VAN.4<br><br>FIA_AFL.1<br><br>FIA_ATD.1(1)<br><br>FIA_ATD.1(2)<br><br>FIA_ATD.1(3)<br><br>FIA_UAU.1<br><br>FIA_UAU.2<br><br>FIA_UAU.5<br><br>FPT_ITA.1<br><br>FPT_TDC.1(1)<br><br>FPT_TDC.1(2) | FIA_UID.2 plays a small role in satisfying this objective by ensuring that every user is identified before the TOE performs any mediated functions. In some cases, the identification cannot be authenticated (e.g., anonymous access by a relying party, in which case the identity is presumed to be authentic). In other cases (e.g., directory administrator, authenticated relying parties), the identity of the user is authenticated. It is impractical to require authentication of all relying parties, therefore the requirements specified require authentication where it is deemed necessary. This does impose some risk that actions taken by an anonymous relying party may not be traceable to a human user.<br><br>FIA_ATD.1 is iterated several times to ensure that the attributes of the different users of the TOE are specified correctly. This requirement is needed because it is here that the attributes that will be used by the TOE in making access control decisions are specified.<br><br>FIA_UAU.1 contributes to this objective by limiting the services and directory objects that are provided by the TOE to unauthenticated users.<br><br>FIA_UAU.2 specifies that all other users of the TOE not covered by FIA_UAU.1 have to authenticate, controlling their access to the TOE such that they cannot perform actions until after authentication is successful.<br><br>The PP requires multiple authentication mechanisms to be available. FIA_UAU.5 requires that these mechanisms be used for the appropriate set of users defined by FIA_ATD.1(*), and also defines the rules for when they are used. It also defines the |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| | | "third-party authentication" that takes place when a request is chained to the TOE, which is another way that users have of logically accessing the TOE.<br><br>Local authentication is required to ensure someone that has physical access to the TOE and has not been granted logical access (e.g., a janitor) cannot gain unauthorized logical access to the TOE. |
| | | The AVA_VAN.4 requirement as applied to the local authentication mechanism. The evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. Penetration testing is performed by the evaluator assuming an attack potential of moderate. This requirement ensures the evaluator has performed an analysis of the authentication mechanism to ensure the probability of guessing a user's authentication data would require a high-attack potential, as defined in Annex B of the CEM.<br><br>FTA_TSE.1 contributes to this objective by limiting a user's ability to logically access the TOE. This requirement provides the ability to control when (e.g., time and day(s) of the week) and where (e.g., from a specific network address) TOE users can access the TOE.<br><br>FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts. This requirement focuses on preventing inappropriate access to the TOE by guessing authentication information, which is why the requirements are worded to cover remote authentication requests. Since relying parties are untrusted with respect to the TOE, all of their authentication attempts are subject to investigation.<br><br>The FTA_SSL family partially satisfies the O.ROBUST_TOE_ACCESS objective by ensuring that user's sessions are afforded some level of protection. FTA_SSL.1 provides the Platform Administrator the capability to specify a time interval of inactivity in which an unattended local administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources. FTA_SSL.2 provides administrators the ability to lock their local administrative session. This component allows administrators to protect their session immediately, rather than waiting for the time-out period and minimizes their session's risk of exposure. FTA_SSL.3 takes into account remote sessions. After an administrator-defined time interval of inactivity remote sessions will be terminated; this includes relying party sessions and remote administrative sessions (both directory sessions and platform sessions). This component is especially necessary, since remote sessions are not typically afforded the same physical protections that local sessions are provided.<br><br>FPT_ITA.1 specifies the ability to control access to TSF data in a manner that makes certificates and revocation lists available for authentication decisions in a timely fashion.<br><br>The two iterations of FPT_TDC are used to specify capabilities of the TOE that are needed when a user is accessing a TOE. FPT_TDC.1(1) is needed in order to interpret timestamps on certificates so that a determination can be made about whether they have expired. FPT_TDC.1(2) is needed so that distinguished names can be interpreted when they are presented to the TOE, and access granted if appropriate. |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| O.NONREPUDIATION<br><br>At the option of an administrator, the TSF must be able to provide non-repudiation services for transmitted and received repository data. These services must include both the generation and verification of evidence for non-repudiation, including a timestamp, and notification that the evidence of receipt the TOE is waiting for is overdue. | FCO_PRA_(EXT).1<br><br>FMT_MOF.1(1) | The objective is met by an extended requirement based on the FCO class.<br><br>FCO_PRA_(EXT).1.1(1) covers the case where the TOE is the originator of the information; in this case, the TOE must timestamp the fact that it was the TOE that initiated a replication event.<br><br>FCO_PRA_(EXT).1.2(1) covers the case where the TOE is the recipient of the information; in this case, the TOE must timestamp the fact that it received replica data in a way that proves the TOE was the one that received it.<br><br>FCO_PRA_(EXT).1.3(1) is the requirement that the originating TOE be capable of associating the evidence from FCO_PRA_(EXT).1.1(1) and FCO_PRA_(EXT).1.2(1) with the data that were replicated; this meets the objective that proves who sent the replica data, who received the replica data, and the time that those events occurred.<br><br>FCO_PRA_(EXT).1.4(1) meets the objective that notification is given when receipt is not acknowledged by requiring the TOE to send the notification to a security administrator.<br><br>FCO_PRA_(EXT).1.5(1) provides the capability for a user to invoke the TSF to provide the non-repudiation evidence for a given set of replica data (usually a CRL). While the other element focus on production and collection of the information, this element is for the on-demand display of the information.<br><br>FMT_MOF.1(1) is used to satisfy the "at the option of the administrator portion of the objective." Through this requirement, the administrator has the ability to specify, on a replica-by-replica basis, whether the TOE generates and maintains the required information or not. |
| O.TRUSTED_PATH<br><br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. | FTP_ITC_(EXT).1(1)<br><br>FTP_ITC_(EXT).1(2)<br><br>FTP_TRP_(EXT).1(1)<br><br>FTP_TRP_(EXT).1(2) | FTP_TRP_(EXT).1.1 requires the TOE to provide a mechanism that creates a distinct communication path that protects the data that traverses this path from disclosure (first iteration) or modification (second iteration). This requirement ensures that the TOE can identify the end points and ensures that a user cannot insert themselves between the user and the TOE, by requiring that the means used for invoking the communication path cannot be intercepted and allow a "man-in-the-middle-attack" (this does not prevent someone from capturing the traffic and replaying it at a later time – see FPT_RPL.1). Since the user invokes the trusted path (FTP_TRP_(EXT).1.2) mechanism they can be assured they are communicating with the TOE. FTP_TRP_(EXT).1.3 mandates that the trusted path be the only means available for providing identification and authentication information, therefore ensuring a user's authentication data will not be compromised when performing authentication functions. Furthermore, the remote administrator's communication path is encrypted during the entire session.<br><br>FTP_ITC_(EXT).1(1) and FTP_ITC_(EXT).1(2) are similar to FTP_TRP_(EXT).1(1) and FTP_TRP_(EXT).1(2), in that they require a mechanism that creates a distinct communication path with the same characteristics, however FTP_ITC_(EXT).1(1) and FTP_ITC_(EXT).1(2) is used to protect communications between IT entities, rather than between a human user and an IT entity. FTP_ITC_(EXT).1.3 requires the TOE to initiate the trusted channel, which ensures that the TOE has established a communication path with an authorized IT entity and not some other entity pretending to be an authorized IT entity. |

| Objective | Requirements Addressing the Objective | Rationale |
|---|---|---|
| O.VULNERABILITY_ANALYSIS_TEST<br><br>The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies. | AVA_VAN.4 | AVA_VAN.4 requires the evaluator to perform a search of public domain sources to identify potential vulnerabilities in the TOE. The evaluator will perform an independent, methodical vulnerability analysis of the OE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE. The evaluator will conduct penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing **Moderate** attack potential. For those vulnerabilities that are not eliminated by the developer, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a moderate attack potential, which is in keeping with the desired assurance level of this TOE. |

## 6.4   RATIONALE FOR ASSURANCE REQUIREMENTS

The EAL definitions and assurance requirements in Part 3 of the CC were reviewed and the *Medium Robustness Assurance Package* as defined in Section 5.3 was believed to best achieve the goal of addressing circumstances where developers and users require a moderate to high level of independently assured security in commercial products. The assurance package selection was based on:

• recommendations documented in the GIG;

• DoD Instruction 8500.1; and

• the postulated threat environment.

This collection of assurance requirements require TOE developers to gain assurance from good software engineering development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. Rationale for individual assurance requirements is provided in Table 6.5.

The Government's guidance in the GIG was consulted and found to also support the chosen assurance package. Specifically, the GIG states that medium robustness security services and mechanisms provide for additional safeguards above the DoD minimum and require good assurance security design as specified in EAL3 or greater.

The postulated threat environment specified in Section 3 of this PP was used in conjunction with the Information Assurance Technical Framework (IATF) Robustness Strategy guidance to derive the chosen assurance level.

These three factors were taken into consideration and the conclusion was that the medium robustness assurance package was the appropriate level of assurance.

## 6.5   RATIONALE FOR DEPENDENCIES

**Each functional requirement, including extended requirements was analyzed to determine that all dependencies were satisfied.  All requirements were then analyzed to determine that no additional dependencies were introduced as a result of completing each operation.  Table 6.3 identifies the functional requirement, and its correspondent dependency.**

Table 6.4 provides the analysis and rationale for dependencies not required in this PP.

In Table 6.3, the "Component" column lists all of the components included in this PP; each one is assigned a unique ID number in the "ID" column.  Each component's dependencies (from the CC) are listed in the "Dependency" column.  The "Satisfied" column indicates how the dependencies are satisfied, with the number referencing the ID number of the component included in the PP that satisfies the dependencies.  N/A is used when there are no dependencies for a component, and a reference to Table 6.4 is included when the dependency is not met but justified in Table 6.4.

**Table 6.3 – Dependencies Table**

| ID | Component | Dependency | Satisfied |
|----|-----------|------------|-----------|
| 1 | FAU_ARP.1 | FAU_SAA.1 | 5 |
| 2 | FAU_ARP_ACK_(EXT).1 | FAU_ARP.1 | 1 |
| 3 | FAU_GEN.1-NIAP-0347 | FPT_STM.1 | 55 |
| 4 | FAU_GEN.2-NIAP-0410 | FAU_GEN.1 | 3 |
|   |  | FIA_UID.1 | 36 (Hierarchical) |
| 5 | FAU_SAA.1-NIAP-0407 | FAU_GEN.1 | 3 |
| 6 | FAU_SAR.1(1) | FAU_GEN.1 | 3 |
| 7 | FAU_SAR.1(2) | FAU_GEN.1 | 3 |
| 8 | FAU_SAR.2 | FAU_SAR.1 | 6, 7 |
| 9 | FAU_SAR.3 | FAU_SAR.1 | 6, 7 |
| 10 | FAU_SEL.1-NIAP-0407 | FAU_GEN.1 | 3 |

| ID | Component | Dependency | Satisfied |
|---|---|---|---|
| | | FMT_MTD.1 | 41 |
| 11 | FAU_STG.1-NIAP-0429 | FAU_GEN.1 | 3 |
| 12 | FAU_STG.3 | FAU_STG.1 | 11 |
| 13 | FAU_STG.NIAP-0414-1-NIAP-0429 | FAU_STG.1 | 11 |
| | | FMT_MTD.1 | 41 |
| 14 | FCO_PRA_(EXT).1(1) | FMT_SMR.1 | 48 |
| | | FPT_STM.1 | 55 |
| | | FDD_RPL_(EXT).1 | 27 |
| | | FTP_ITC.1 | 68, 69 |
| 16 | FCS_BCM_(EXT).1 | None | N/A |
| 17 | Reserved | | |
| 18 | FCS_CKM.1(1) Symmetric keys  FCS_CKM.1(2) Asymmetric keys | FCS_CKM.4, FCS_COP.1 FCS_CKM.2 FMT_MSA.2 | 20 21-24 19 N/A – See Table 6.4 below. |
| 19 | FCS_CKM.2 | FCS_CKM.1 FCS_CKM.4 FMT_MSA.2 | 18 20 N/A – See Table 6.4 below. |
| 20 | FCS_CKM.4 | FDP_ITC.1 or FCS_CKM.1 | 18 |
| | | FMT_MSA.2 | N/A – See Table 6.4 below. |
| 21 | FCS_COP.1(1) encryption/decryption | FCS_BCM_(EXT).1 | 16 |
| | | FCS_CKM.1(1) Symmetric keys | 18 |
| | | FCS_CKM.1(2) Asymmetric keys | 18 |
| | | FCS_CKM.4 | 20 |
| 22 | FCS_COP.1(2) cryptographic signature | FCS_BCM_(EXT).1 | 16 |
| | | FCS_CKM.1(1) Symmetric keys | 17 |

| ID | Component | Dependency | Satisfied |
|---|---|---|---|
| | | FCS_CKM.1(2) Asymmetric keys | |
| | | FCS_CKM.4 | 20 |
| 23 | FCS_COP.1(3) cryptographic hashing | FCS_BCM_(EXT).1 | 16 |
| 24 | FCS_COP.1(4) cryptographic key agreement | FCS_BCM_(EXT).1 | 16 |
| | FCS_COP_(EXT).1 Random number generator | FCS_BCM_(EXT).1 | 16 |
| 25 | FDP_ACC.2 | FDP_ACF.1 | 26 |
| 26 | FDP_ACF.1 | FDP_ACC.1 | 25 |
| | | FMT_MSA.3 | N/A – See Table 6.4 below. |
| 27 | FDD_RPL_(EXT).1 | None | N/A |
| 28 | FDP_RIP.2 | None | N/A |
| 29 | FIA_AFL.1 | FIA_UAU.1 | 33 |
| 30 | FIA_ATD.1(1) | None | N/A |
| 31 | FIA_ATD.1(2) | None | N/A |
| 32 | FIA_ATD.1(3) | None | N/A |
| 33 | FIA_UAU.1 | FIA_UID.1 | 36 (Hierarchical) |
| 34 | FIA_UAU.2 | FIA_UID.1 | 36 (Hierarchical) |
| 35 | FIA_UAU.5 | No Dependencies | N/A |
| 36 | FIA_UID.2 | No Dependencies | N/A |
| 37 | FIA_USB.1 | FIA_ATD.1 | 30, 31, 32 |
| 38 | FMT_MOF.1(1) | FMT_SMR.1 | 49 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 39 | FMT_MOF.1(2) | FMT_SMR.1 | 49 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 40 | FMT_MSA.1 | [FDP_ACC.1 or FDP_IFC.1] | 25 |
| | | FMT_SMR.1 | 49 (Hierarchical) |

| ID | Component | Dependency | Satisfied |
|----|-----------|------------|-----------|
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 41 | FMT_MTD.1(1) | FMT_SMR.1 | 49 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 42 | FMT_MTD.1(2) | FMT_SMR.1 | 48 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 43 | FMT_MTD.1(3) | FMT_SMR.1 | 49 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 44 | FMT_MTD.1(4) | FMT_SMR.1 | 49 (Hierarchical) |
| | | FMT_SMF.1 | N/A – See Table 6.4 below. |
| 45 | FMT_MTD.2(1) | FMT_MTD.1 | 41 |
| | | FMT_SMR.1 | 49 (Hierarchical) |
| 46 | FMT_MTD.2(2) | FMT_MTD.1 | 41 |
| | | FMT_SMR.1 | 49 (Hierarchical) |
| 47 | FMT_SMF.1 | None | N/A |
| 48 | FMT_SMR.2(1) | FIA_UID.1 | 36 (Hierarchical) |
| 49 | FMT_SMR.2(2) | FIA_UID.1 | 36 (Hierarchical) |
| 50 | FPT_ITA.1 | None | N/A |
| 51 | FPT_RCV.2 | FPT_TST.1 | 58, 59 |
| | | AGD_OPE.1 | Medium Robust Assurance |
| 52 | FPT_RPL.1 | None | N/A |
| 53 | Reserved | | |
| 54 | reserved | | |
| 55 | FPT_STM.1 | None | N/A |
| 56 | FPT_TDC.1(1) | None | N/A |

| ID | Component | Dependency | Satisfied |
|---|---|---|---|
| 57 | FPT_TDC.1(2) | None | N/A |
| 58 | FPT_TST_(EXT).1 | None | |
| 59 | FPT_TST.1(1) FPT_TST.1(2) | FCS_COP.1(1)-(4) | 21-24 |
| 60 | FRU_RSA.1(1) | None | N/A |
| 61 | FRU_RSA.1(2) | None | N/A |
| 62 | FTA_SSL.1 | FIA_UAU.1 | 33 |
| 63 | FTA_SSL.2 | FIA_UAU.1 | 33 |
| 64 | FTA_SSL.3(1) | None | N/A |
| 65 | FTA_SSL.3(2) | None | N/A |
| 66 | FTA_TAB.1 | None | N/A |
| 67 | FTA_TSE.1 | None | N/A |
| 68 | FTP_ITC_(EXT).1(1) | None | N/A |
| 69 | FTP_ITC_(EXT).1(2) | None | N/A |
| 70 | FTP_TRP_(EXT).1(1) | None | N/A |
| 71 | FTP_TRP_(EXT).1(2) | None | N/A |

**Table 6.4 – Unsupported Dependency Rationale**

| Requirement | Dependency | Dependency Analysis and Rationale |
|---|---|---|
| FCS_CKM.1 FCS_CKM.4 | FMT_MSA.2 | This dependency is not applicable for this TOE since it's redundant to the requirements specified in the FCS components. |
| FDP_ACF.1 | FMT_MSA.3 | This dependency is not applicable for this TOE since restrictive default values for the SFP is already required in FDP_ACF.1, and this PP does not want to allow the default to be changed. |
| FMT_MOF.1(*) FMT_MSA.1 FMT_MTD.1(*) | FMT_SMF.1 | This dependency is not applicable for this TOE since all the management functions required by the TOE are implicit in the other FMT components. FMT_SMF.1 is only used to specify the backup, recovery and archive requirements. |

### *6.6   RATIONALE FOR EXTENDED REQUIREMENTS*

Table 6.5 presents the rationale for the inclusion of the extended requirements found in this PP.

**Table 6.5 – Rationale for Extended Requirements**

| Extended Requirement | Identifier | Rationale |
|---|---|---|
| FAU_ARP_ACK_DIR_(EXT).1 | Security alarm acknowledgement for Directory | This extended requirement is necessary since a CC requirement does not exist to ensure an administrator will be aware of the alarm. The intent is to ensure that if an administrator is logged in and not physically at the console or remote workstation the message will remain displayed until the administrators have acknowledged it. The message will not be scrolled off the screen due to other activity-taking place (e.g., the auditor is running an audit report). The following are the dependencies for this component:  FAU_ARP.1. |
| FAU_STG.NIAP-0429 | Site-Configurable Prevention of Audit Loss | This extended requirement is taken from the NIAP interpretation (originally I-0414 and subsequently modified by I-0429) to require functionality that is not available with current CC requirements. The authors of this PP want to provide the Security Administrator with the option of what action to take when the audit trail is full. The following are the dependencies for this component:  FAU_STG.1, FMT_MTD.1. |

| Extended Requirement | Identifier | Rationale |
|---|---|---|
| FCO_PRA_(EXT).1 | Proof of Replication Activity | This extended requirement is necessary since the existing CC non-repudiation components cover the case where the non-repudiation is required at the request of an originator or recipient. For compliant TOEs, the functionality is that communication is generated by the TOE (and a peer IT entity), but the proof is requested by an administrator or other designated personnel. This model is not supported by the existing CC requirements, so extended requirements are needed. Additionally, the CC requirements make no provision for notification that the information was not received (which provides a portion of the non-repudiation evidence required), which again is specified through the use of an extended requirement.<br><br>The following are the dependencies for this component: FMT_SMR (to specify the roles that are allowed to configure the service, and to receive notification if receipt is not acknowledged), FPT_STM.1 (to timestamp the evidence of origin or receipt), FDD_RPL_(EXT).1 (the replication mechanism that this requirement provides evidence relating to), and FTP_ITC.1 (to provide the trusted channel needed to perform the replication activity, and to transmit the proof of receipt). |
| FCS_BCM_(EXT).1 | Baseline cryptographic module | This extended requirement is necessary since the CC does not provide a means to specify a cryptographic baseline of implementation.<br><br>The following are the dependencies for this component: none. |
| FCS_COP_(EXT).1 | Cryptographic Operation (Random Number Generation) | This extended requirement is necessary since the CC cryptographic operation components are focused on specific algorithm types and operations requiring specific key sizes.<br><br>The following are the dependencies for this component: FCS_BCM_(EXT).1 |

| Extended Requirement | Identifier | Rationale |
|---|---|---|
| FDD_RPL_(EXT).1 | Replication of directory data with security attributes. | This extended component is necessary to specify a unique requirement for a technology specific security service that is not addressed by the CC.   This service is required to meet O.REPLICATION.<br><br>The following are the dependencies for this component:  None. |
| FPT_TST_(EXT).1 | TSF testing | This extended component is necessary to specify the self-testing functionality required for medium robustness.<br><br>The following are the dependencies for this component:  FCS_COP.1. |
| FTP_ITC_(EXT).1 | Inter-TSF trusted channel | This extended component is necessary because it removes a contradiction from the requirement.  Per OD-232 an interpretation is being created to fix the contradiction.  When the final international interpretation is created compliant TOEs should use the updated CC requirements rather than the extended requirements. |
| FTP_TRP_(EXT).1 | Trusted Path | Same as FTP_ITC_(EXT).1. |
| AVA_CCA_(EXT).2 | Systematic Cryptographic Module Covert Channel Analysis | This extended requirement is necessary since the CC does not have requirements to perform a covert channel analysis on information that does not have an information flow control policy. This requirement ensures that the bandwidth of critical security parameters (e.g., keys) associated with the cryptographic module is documented. |

{This page intentionally left blank}

# 7 ACRONYMS

**Table 7.1 – List of Acronyms**

| | |
|---|---|
| **ACIA** | Access Control Inner Administrative Area |
| **ACIP** | Access Control Inner Point |
| **ACI** | Access Control Information |
| **ACL** | Access Control List |
| **ACSA** | Access Control Specific Area |
| **ACSP** | Access Control Specific Point |
| **ADS** | Authoritative Data Source |
| **ADUA** | Administrative Directory User Agent |
| **AM** | Assurance Maintenance |
| **ANSI** | American National Standards Institute |
| **ARL** | Authority Revocation List |
| **C/S/A** | CINC/Service/Agency |
| **CA** | Certificate Authority |
| **CC** | Common Criteria |
| **CIMC** | Certificate Issuing and Management Component |
| **CINC** | Commander-in-Chief |
| **CM** | Configuration Management |
| **CMA** | Certificate Management Authority |
| **DA** | Directory Administrator |
| **DACD** | Directory Access Control Domains |
| **DAP** | Directory Access Protocol |
| **DES** | Data Encryption Standard |

| DIB | Directory Information Base |
|---|---|
| DISA | Defense Information Services Agency |
| DIT | Directory Information Tree |
| DN | Distinguished Name |
| DoD | Department of Defense |
| DSA | Directory Service Agent |
| DSP | Directory System Protocol |
| DUA | Directory User Agent |
| EAL | Evaluation Assurance Level |
| EDI_PI | Electronic Data Interchange Personnel Identifier |
| FIPS | Federal Information Processing Standard |
| FOUO | For Official Use Only |
| FTP | File Transfer Protocol |
| GDS | Global Directory Service |
| GIG | Global Information Grid |
| HAG | High Assurance Guard |
| HTTP | Hypertext Transport Protocol |
| I&A | Identification and Authentication |
| ICMP | Internet Control Message Protocol |
| ID | Identification |
| IP | Internet Protocol |
| IT | Information Technology |
| KEA | Key Exchange Algorithm |
| KM | KMI Manager |
| KMI | Key Management Infrastructure |

| KR | Key Recovery |
|---|---|
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| MD | Misuse Detection System |
| NIAP | National Information Assurance Partnership |
| NIC | Network Interface Card |
| NIPRNet | Non-classified Internet Protocol Router Network |
| NIST | National Institute of Standards and Tests |
| NSA | National Security Agency |
| NTP | Network Time Protocol |
| PKCS | Public Key Cryptography Standard |
| PKI | Public Key Infrastructure |
| PP | Protection Profile |
| PRSN | Primary Services Node |
| PSN | Product Source Node |
| PUB | Publication |
| RFC | Request for Comments |
| RI | Repository Information |
| RL | Revocation List |
| RM | User Registration Manager |
| SA | System Administrator |
| SASL | Simple Authentication and Security Layer |
| SFP | Security Function Policy |
| SIPRNet | Secret Internet Protocol Router Network |
| | |

| SMTP | Simple Message Transfer Protocol |
|------|----------------------------------|
| SSL  | Secure Socket Layer |
| SSO  | System Security Officer |
| ST   | Security Target |
| TCP  | Transmission Control Protocol |
| TLS  | Transport Layer Security |
| TOE  | Target of Evaluation |
| TP   | Trusted Path |
| TSC  | TSF Scope of Control |
| TSF  | TOE Security Functions |
| TSFI | TSF Interface |
| TSP  | TOE Security Policy |
| UDP  | User Datagram Protocol |
| UTC  | Coordinated Universal Time |
| VPN  | Virtual Private Network |

{This page intentionally left blank}

# 8  REFERENCES

## *8.1  DIRECTORY REFERENCES*

1) Basic and Simplified Access Control in LDAP *<draft-legg-ldap-acm-bac-01.txt>*, Legg, S; September 2002.

2) Chadwick, David W., *Understanding X.500 – The Directory*, 1994.

3) Cheresh, Beth and Doug Cheresh, *Understanding Directory Services*, New Riders Publishing, 2000.

4) *Common Criteria for Information Technology Security Evaluation*, Version 2.1, CCIMB-99-031 (ISO/IEC 15408:1999), August 1999

5) Descriptions of SHA-256, SHA-384, and SHA-512

6) *Digital Signatures using Reversible Public Key (rDSA),* ANSI X9.31-1998

7) *Guide for the Production of Protection Profiles and Security Targets*, 2001-01-04, ISO/IEC PDTR 15446.

8) Housley and Polk, *Internet X.509 Public Key Infrastructure: Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates*, March 1999.  (www.ietf.org/rfc/rfc2528.txt)

9) Housley, Ford, Polk, and Solo, *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile,* January 1999.  (www.ietf.org/rfc/2459.txt)

10) Howes, Timothy, Mark Smith, and Gordon Good, *Understanding and Deploying LDAP Directory Services*, New Riders Publishing, 1999.

11) International Standard ISO 10181-3 *Access Control Framework*

12) ITU-T Recommendation X.501 (1997): Information Technology - Open Systems Interconnection – The Directory: Models, 1997.

13) ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection – The Directory: Authentication Framework, 1997.

14) ITU-T Recommendation X.511 (1997): Information Technology - Open Systems Interconnection – The Directory: Abstract Services Definition, 1997.

15) ITU-T Recommendation X.521 (1997): Information Technology - Open Systems Interconnection – The Directory: Selected Object Classes, 1997.

16) *Key Agreement and Key Transport Using Elliptic Curve Cryptography*, X9.63

17) Krawczy, Bellar, and Canett, *HMAC: Keyed-Hashing for Message Authentication*, February 1997.

18) National Institute of Standards and Technology, *Computer Data Authentication*, Federal Information Processing Standards Publication 113, 30 May 1985.

19) National Institute of Standards and Technology, *Data Encryption Standard*, Federal Information Processing Standards Publication 46-3, 25 October 1999.

20) National Institute of Standards and Technology, *DES Modes of Operation*, Federal Information Processing Standards Publication 81, 2 December 1980.

21) National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-2, 27 January 2000.

22) National Institute of Standards and Technology, *Secure Hash Standard*, Federal Information Processing Standards Publication 180-1, 17 April 1995.

23) National Institute of Standards and Technology, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards Publication 140-2, 25 May 2001.

24) National Security Agency, Strong Authentication for X.500, Revision B, 12 May 1999.

25) National Security Agency, X.509 Certificate and Certificate Revocation List Profiles and Certification Path Processing Rules for MISSI, Revision D, 12 May 1999.

26) *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, X9.42-2001.

27) RSA Cryptography Standard, RSA Laboratories, PKCS #1, v2.0 1 October 1998

28) Signed Directory Operations using S/MIME *<Draft_ietf_ldapext_sigops_03.txt>*, Greenblatt, Bruce and Richard, Pat; July 1998.

29) Skipjack and KEA Algorithm Specifications, Version 2.0, 29 May 1998.

30) *The Elliptic Curve Digital Signature Algorithm,* ANSI X9.62-1998

31) *Triple DES Encryption Algorithm Modes of Operation,* ANSI X9.52-1998

## 8.2 REQUIREMENTS REFERENCES

1) *Department of Defense Key Management Infrastructure / Public Key Infrastructure Capability Increment 1 Overview*, Version 2.0, 25 May 2000.

2) *Firewall Recommendation Report for the GDS Directory*, 8 December 2000

3) *GDS Baseline Requirements for JDSWG 4-20-011.xls*

4) GDS/KMI Interface Control Doc v.1, GDS KMI ICD v1.0.doc

5) *Global Directory Service Backup & Failover Plan*, 12 January 2001

6) *Global Directory Service Roadmap*, October 2000

7) *Global Directory Service System Architecture Version1.0*, 26 February 2001

8) *Key Management Infrastructure (KMI): Capability Increment 1: System Interface Description*, 13 April 2000.

9) *KMI CI-1 CONOPS, Directory Section*, 30 June 2000.

10) *KMI Directory Schema*, (CI-1 Final), Version 2.1, 5 March 2001.

11) *KMI-DISA Global Directory Service: Service Level Agreement*, V0.3, 17 November 2000.

12) *KMI Security Architecture for Capability Increment 1 (CI-1)*, version 1, 1 December 2000.

13) KMI Security Policy and Requirements, 6 September 2000.

14) *KMI Security Policy for Capability Increment 1 (CI-1)*, Version 0.2, 22 September 2000.

15) *Consistency Instruction Manual For Development of US Government Protection Profiles (PP) For use in Medium Robustness Environments*, NIAP Protection Profile Review Board, Release 2.0, 1 March 2004.

16) *NSA Security Guidance for DoD Class 4 PKI Directory Service (DS)*, 15 June 2000.

17) *Operational Requirements for the Defense Message System Directory Services*, Allied Communications Protocol (ACP) 120, Version 1.0, 17 December 1999.

18) *Public Key Infrastructure Implementation Plan for the Department of Defense*, Version 3.1, 18 December 2000.

19) *Public Key Infrastructure Roadmap for the Department of Defense*, version 5.0, 18 December 2000.

20) *System Description for Capability Increment 1 (CI-1),* Revision 2.2, 28 February 2001.

21) *System Requirements Specification (SRS) for Capability Increment 1 (CI-1)*, Revision 2.2, 28 February 2001.

22) *X.509 Certificate Policy for the US Department of Defense*, Version 5.2, 13 November 2000.

## *8.3 RELATED PROTECTION PROFILES*

1) *A Goal VPN PP for Protecting Sensitive Information,* Validated version to be determined.

2) *Certificate Issuing and Management Components Family of Protection Profiles*, Draft Version 1.0, 31 October 2001.

3) *Department of Defense Public Key Infrastructure Target Token Protection Profile*, Validated version to be determined.

4) *Final U.S. Department of Defense Traffic-Filter Firewall Protection Profile for Medium Robustness*, Version 1.4, 1 May 2000.

5) *Intrusion Detection System Analyzer Protection Profile*, Validated version to be determined.

6) *Intrusion Detection System Protection Profile*, Validated version to be determined.

7) *Intrusion Detection System Scanner Protection Profile*, Validated version to be determined.

8) *Intrusion Detection System Sensor Protection Profile*, Validated version to be determined.

9) *Protection Profile for Single Level Operating Systems in Environments Requiring Medium Robustness*, Version 1.22, 23 May 2001.

10) *U.S. Department of Defense Application Firewall Protection Profile for Medium Robustness*, Version 1.0, dated 28 June 2000.

11) *U.S. Department of Defense Remote Access Protection Profile for High Assurance Environments*, Validated version to be determined.

12) *Web Browser Protection Profile*, Validated version to be determined.

13) *Web Server Protection Profile*, Validated version to be determined.

# 9   TERMINOLOGY

***3rd Party Introduction*** — An example of a *distributed authentication* mechanism.  A form of authentication used in the chaining process when a TOE trusts that the peer trusted directory correctly verified the authentication credentials of the relying party before passing the chained request to the TOE.

***3rd Party Presentation*** — An example of a *distributed authentication* mechanism.  A form of authentication used in the chaining process when a TOE trusts that the peer directory ensured the integrity and, if necessary, the confidentiality of the authentication credentials passed to the TOE as part of the chained request.

***Access*** — Interaction between an entity and an object that results in the flow or modification of data.

***Access Control*** — Security service that controls the use of resources[1] and the disclosure and modification of data.[2]

***Access Control Information (ACI)*** — Information stored in the directory that is used to determine which users have been granted access to directory objects and what type of access has been granted (e.g., read, write).

***Access Control Decision Function*** — A specialized function that makes access control decisions by applying access control policy rules to an access request.

***Access Control Domain*** —  the repository information in a single Directory server can be split up into arbitrary overlapping collections of entries to which a uniform application of an access control policy can be applied.  Each of these groupings is referred to as an Access Control Domain.

***Access Control Scheme*** — Access control scheme, from X.500, identifies the access control model and access control decision functions.  Examples of access control schemes include X.500 Basic Access Control with role-base and X.500 Simple Access Control with role-based.

***Accountability*** — Property that allows activities in an IT system to be traced to the entity responsible for the activity.

***Administrative Directory User Agent (ADUA)*** — A specialized trusted user interface to perform administrative functions on the directory.

***Administrator*** — A user who has been specifically granted the authority to manage the TOE or a subset of the TOE, and whose actions may affect the TSP.  Administrators may possess special privileges that provide capabilities to override portions of the TSP.

***Anonymous Relying Party*** — Anonymously authenticated relying party.

---

[1] Hardware and Software

[2] Stored or communicated.

*Application Note* — Supporting information that is considered relevant or useful for the construction, evaluation, or use of the TOE.

*Assurance* — A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

*Asymmetric Cryptographic System* — A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

*Asymmetric Key* — The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.

*Attack* — An intentional act attempting to violate the security policy of an IT system.

*Attack Potential* — The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

*Attribute* — A property that is associated with an entry. Attributes may be of a user type or operational type. User attributes are those attributes accessible by users. Operational attributes are attributes used by the directory and not accessible by users. An attribute is made up of attribute values and attribute type. The attribute type defines how the attribute value is used and processed. Attributes may be mandatory or optional.

*Audit* — To conduct an internal or independent review and assessment of records and/or activities.

*Auditor* — Role required by the TOE for a type of Administrative user that is given privileges commensurate with performing audit functions.

*Authentication* — Security measure that verifies a claimed identity.

*Authentication Data* — Information used to verify a claimed identity.

Authority Revocation List — See Revocation List.

*Authorization* — Permission, granted by an entity authorized to do so, to perform functions and access data.

*Authorized User* — An authenticated user who may, in accordance with the TSP, perform an operation.

*Availability* — Timely[3], reliable access to IT resources.

*Basic Access Control* — One of three X.500-defined access control schemes for the directory. It is defined in 1997 version of X.501.

---

[3] According to a defined metric.

***Black Box*** — An abstraction of a device or system in which only its externally visible behaviour is considered and not its implementation or "inner workings".

***Bind*** — The protocol used to connect to a directory.

***Certification Authority (CA)*** — An entity authorized to issue, manage, and revoke certificates.

***Certificate-based authentication (two-way)*** — Identification and authentication is bi-directional, both entities provide proof of identity before the authentication is considered complete.

Certificate Revocation List (CRL) — See Revocation List.

***Chaining*** — Process used in a distributed directory environment in which a query for information is passed from one DSA to another. The results of the query are then returned to the originating DSA, which is then returned to the client. There are two authentication mechanisms used in the chaining process that ensure the access control policies can apply to these requests: "3rd party introduction" and "3rd party presentation".

***Common Criteria*** — The Common Criteria represents the outcome of a series of efforts to develop criteria for evaluation of IT security that are broadly useful within the international community.

***Compromise*** — Violation of a security policy.

***Confidentiality*** — A security policy pertaining to disclosure of data.

***Connectivity*** — The property of the TOE that allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

***Console*** — A combination of keyboard and screen connected to an operating system port specified for administrator access. Historically this was limited to a hard-wired character-only terminal connected to a serial port.

***Critical Security Parameters (CSP)*** — Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

***Cryptographic Administrator*** — An authorized user role that has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

***Cryptographic Algorithm*** — Asymmetric: A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

***Cryptographic Algorithm*** — Symmetric: A cryptographic algorithm that uses a single, secret key for both encryption and decryption.

***Cryptographic Boundary*** — An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

***Cryptographic Key (key)*** — A parameter used in conjunction with a cryptographic algorithm that determines:

the transformation of plaintext data into ciphertext data,

the transformation of cipher text data into plaintext data,

a digital signature computed from data,

the verification of a digital signature computed from data, or

a digital authentication code computed from data.

***Cryptographic Module (cryptomodule)*** — The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

***Cryptographic Module Security Policy*** — A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

***Data Manager*** — A role required by the TOE for trusted human users or external IT entities responsible for providing or accessing a set of trusted data (TSF data).

***Defense-in-Depth (DID)*** — A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

***Dependency*** — A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.

***Digital Certificate*** — An element of a PKI that is used to bind a key to an entity. There are many types of digital certificates resulting from differing standards and operational environments. For the purposes of this PP, "digital certificate" should be generically.

***Digital Signature*** — A non-forgeable transformation of data that allows proof of the source and verification of the integrity of that data.

***Directory*** — A repository, centralized or distributed in nature, from which known system entities may obtain public key certificates, or other information.

***Directory Access Control Domain (DACD)*** — The scope of an access control policy.

***Directory Administrator (DA)*** — Role supported by the TOE that is given privileges commensurate with administering the TOE.

***Directory Information Base (DIB)*** — A term frequently used to define the *repository information.* The complete set of all the information held in the directory, i.e., the DIB entries and DIB attributes.

***DIB Attribute*** — Each piece of information that describes some aspect of a DIB entry.

***DIB Entry*** — Structures that hold the DIB information, including the objects and its attributes.

***Directory Information Tree*** — Logical structure of information.  Entries of the repository are arranged in the form of a tree known as the Directory Information Tree (DIT) where the vertices represent the RI Entries.

***Directory System Agent (DSA)*** — Term describing the server component of a directory service.   More technically, a DSA is a software process that is responsible for serving all requests (search, read, modify, etc.) to a defined naming context.

***Directory User Agent (DUA)*** — Client application used to access the directory.  More technically, a DUA is a software application that communicates with a DSA to issue requests (search, read, modify, etc.).

***Discretionary Access Control (DAC)*** — A means of restricting access to objects based on the identity of subjects and/or groups to which they belong.  These controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

***Distributed Authentication*** — An authentication mechanism used in a distributed directory that may allow the authentication data, the I&A mechanism, and the repository information being accessed to reside on separate servers.

***Distributed Directory*** — A directory system that comprises multiple individual directory servers that interoperate to form an overall distributed directory that receives its data from various sources, protects it in accordance with the system security policy, and makes it available in accordance with the system security policy.

***Distinguished Name*** — A representation of a directory name, defined as a construct that identifies a particular object from among the set of all objects.

***Enclave*** — A collection of entities under the control of a single authority and having a homogeneous security policy.  They may be logical, or may be based on physical location and proximity.

***Encrypted Channel*** — A communications channel connecting the TOE to an outside IT entity that has been secured to prevent disclosure of information in the channel.

***Entity*** — A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

***Evaluation Assurance Level (EAL)*** — A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.

***External IT entity*** — Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

***Global Directory Service (GDS)*** — An integrated enterprise level directory service that facilitates sharing of information from various data sources.

*Human User* — Any person who interacts with the TOE.

*Intrusion Detection System (IDS)* — An example of a trusted external IT entity that identifies events that that may be indicative of an attack on a system. There are various types of IDS including network based IDS, platform based IDS, etc.

*Internet Engineering Task Force (IETF)* — Open international community concerned with the evolution of the Internet architecture technologies.

*Identity* — A representation (e.g. a string) uniquely identifying an authorized user, which can be either the full or abbreviated name of that user or a pseudonym.

*Integrity* — A security policy pertaining to the corruption of data and TSF mechanisms.

*Key Management* — The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs, passwords) during the entire life cycle of the keys, including their generation, storage, distribution, entry and use, deletion or destruction, and archiving.

*Lightweight Directory Access Protocol (LDAP)* — Internet protocol for accessing distributed directory services that act in accordance with X.500 data and service models.

*Named Object[4]* — An object that exhibits all of the following characteristics:

The object may be used to transfer information between subjects of differing user identities within the TSF.

Subjects in the TOE must be able to request a specific instance of the object.

The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

(Note: Due to the deletion of the last sentence in the OS PP (pertaining to intended use of the object being for sharing user data), something may need to be done to the requirements section of the PP (i.e., FDP_ACF) to ensure that some objects, which may satisfy the above but which are not intended for sharing user data do not need a full DAC implementation but rather it is acceptable if they are "owner only" or some other appropriate mechanism).

*Non-Repudiation* — A security policy pertaining to providing one or more of the following:

To the sender of data, proof of delivery to the intended recipient,

To the recipient of data, proof of the identity of the user who sent the data.

*Object* — An entity within the TSC that contains or receives information and upon which subjects perform operations. Examples include a RI entry, attribute, or object class.

---

[4]The only named objects in this PP, are operating system controlled files.

***Operating Environment*** — The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

***Organizational Security Policies*** — One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

***Package*** — A reusable set of either functional or assurance components (e.g. an EAL), combined together to satisfy a set of identified security objectives.

***Password*** — A string of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorization.

***Peer Trusted Directory*** — A trusted external IT entity that performs directory functions as part of a distributed directory system.

***Peer TOEs*** — A Peer Trusted Directory that is also compliant to this PP.

***Platform*** — Typically a device that includes the hardware and software elements that support all or part of the functional requirements of the TOE applications.

***Precedence Levels*** — Predetermined levels of importance used in access control decisions.

***Product*** — A package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.

***Protected Items*** — Data in the TOE that is protected using access control mechanisms.

***Protection Profile (PP)*** — An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

***Public Key Infrastructure (PKI)*** — A mechanism that allows users to securely exchange data through the use of a public and a private cryptographic key pairs that are obtained and shared through a trusted authority.

***Pull Operation*** — An operation in which data is taken as opposed to requested.

***Referral*** — Process used in a distributed directory environment in which a query for information is returned to the client unanswered or partially answered, but with a list of recommended alternate directory servers for the client to query. It is then up to the client to query those additional servers.

***Refinement*** — The addition of details to a component.

***Relying Party*** — Untrusted users or untrusted external IT entities that rely on information in a directory and the integrity of that information in the directory.

***Remote Trusted User*** — A trusted user or trusted external IT entity that accesses the directory from a location outside the boundary of the TOE.

***Replay*** — An attack in which a third party captures a command in transmission and replays it at a later time.

*Replica* — All or a portion of the repository information that is replicated into or out of a directory.

*Replication* — Process used in a distributed directory environment in which a replica is distributed to and/or from other directories.

*Replication Supplier* — A directory that serves as the source of a replica.

*Replication Consumer* — A directory server that serves as the recipient of the replica.

*Repository Data* — A term used to refer to the constituent elements of the repository information for some technical contexts.

*Repository Information  (RI)* — A general term defining the information contained in the directory for use by relying parties.  The repository information is frequently referred to as the Directory Information Base (DIB).

*Revocation List* — A document maintained and published by a certification authority (CA) that lists certificates issued by the CA that are no longer valid.  There are many types of revocation lists including certificate revocation lists (CRL) authority revocation lists (ARL), etc.

*Robustness* — A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly.  DoD has three levels of robustness:

> **Basic:**  Security services and mechanisms that equate to good commercial practices.

> **Medium:**  Security services and mechanisms that provide for layering of additional safeguards above good commercial practices.  **High:**  Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

*Role* — A predefined set of rules establishing the allowed interactions between a user and the TOE.

*Secret* — Information that must be known only to authorized users and/or the TSF in order to enforce a specific SFP.

*Secure State* — Condition in which all TOE security policies are enforced.

*Security Administrator* — Role supported by the TOE, which is a type of Administrative user that is given privileges commensurate with maintaining the security-related functionality of the TOE.  Security Administrators may be responsible for security functions on both the platform and the directory.

*Security attribute* — TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.

*Security Policy* — A precise specification of the security rules under which the TOE shall operate, including the rules derived from the requirements of this document and additional rules imposed by the vendor.

*Security Target (ST)* — A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

*Selection* — The specification of one or more items from a list in a component.

*Subject* — An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

*Subtree* — Grouped set of entries that are administered by the same administrator. Multiple subtrees may exist in a single RI.

*Symmetric key* — A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

*System* — A specific IT installation, with a particular purpose and operational environment.

*Target of Evaluation (TOE)* — An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

*Threat* — Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

*Threat Agent* — Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

*Time stamp* — Electronic seal including a time and/or date indication applied over data.

*Time synchronization System* — An example of a trusted external IT entity that the TOE relies on as a reliable time source.

*TOE resource* — Anything useable or consumable in the TOE.

*TOE Security Functions (TSF)* — A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

*TOE Security Functions Interface (TSFI)* — A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.

*TOE Security Policy (TSP)* — A set of rules that regulate how assets are managed, protected and distributed within a TOE.

*Trusted* — Used to describe any user or IT entity that is authenticated to the TOE with some level of assurance.

*Trusted channel* — A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.

*Trusted path* — A means by which a user and a TSF can communicate with necessary confidence to support the TSP. A mechanism by which a trusted user can communicate directly and reliably with the directory and that can only be activated by the user and cannot be imitated by untrusted software.

***TSF data*** — Data created by and for the TOE that might affect the operation of the TOE.

***TSF Scope of Control (TSC)*** — The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

***Unit of Replication*** — The set of entries and attributes that are specified to be replicated, frequently denoted by the DN at the top of a subtree.

***User*** — Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

***User Class*** — A schema used for determining the rules to be applied to a relying party when deciding the users permissions to the requested protected item (access control decision). Users can be granted permissions based on their distinguished name, identity, subtree information, etc.

***User Data*** — Data created by and for the user that does not affect the operation of the TSF.

***User Group*** — Group that further identifies users in a system.

***Vulnerability*** — A weakness that can be exploited to violate the TOE security policy.

***X.500*** — Set of ISO/ITU specifications defining a distributed directory service.

# APPENDIX A:  STATISTICAL RANDOM NUMBER GENERATOR TESTS

1    A cryptographic module employing random number generators (RNGs) shall perform the following statistical tests for randomness. A single bit stream of 20,000 consecutive bits of output from each RNG shall be subjected to the following four tests: monobit test, poker test, runs test, and long runs test. (These four tests are simply those that formerly existed as the statistical RNG tests in Federal Information Processing Standard 140-2. However, for purposes of meeting this protection profile, these tests must be performed at the frequency specified earlier in this protection profile.)

<p align="center">The Monobit Test:</p>

1. Count the number of ones in the 20,000 bit stream. Denote this quantity by X.

2. The test is passed if $9{,}725 < X < 10{,}275$.

<p align="center">The Poker Test:</p>

1. Divide the 20,000 bit stream into 5,000 contiguous 4 bit segments. Count and store the number of occurrences of the 16 possible 4 bit values. Denote $f(i)$ as the number of each 4 bit value $i$, where $0 < i < 15$.

2. Evaluate the following:

$$X = (16 / 5000) * \left( \sum_{i=0} [f(i)]^2 \right) - 5000$$

3. The test is passed if $2.16 < X < 46.17$.

<p align="center">The Runs Test:</p>

1. A run is defined as a maximal sequence of consecutive bits of either all ones or all zeros that is part of the 20,000 bit sample stream. The incidences of runs (for both consecutive zeros and consecutive ones) of all lengths ($> 1$) in the sample stream should be counted and stored.

2. The test is passed if the runs that occur (of lengths 1 through 6) are each within the corresponding interval specified in the table below. This must hold for both the zeros and ones (i.e., all 12 counts must lie in the specified interval). For the purposes of this test, runs of greater than 6 are considered to be of length 6.

<p align="center">**Table C.1 - Required Intervals for Length of Runs Test**</p>

| Length of Run | Required Interval |
|---|---|
| 1 | 2343 - 2657 |
| 2 | 1135 - 1365 |
| 3 | 542 - 708 |
| 4 | 251 - 373 |

| 5 | 111 - 201 |
|---|---|
| 6 and greater | 111 - 201 |

The Long Runs <u>Test</u>:

1. A long run is defined to be a run of length 26 or more (of either zeros or ones).
2. On the sample of 20,000 bits, the test is passed if there are no long runs.

---

**i** A deletion of CC text was performed in FPT_TST.1.1(1). Rationale: The word "TSF" was deleted to allow for the demonstration of the correct operation of a number of cryptographic related self tests.

> FPT_TST.1.1(1) **Refinement:** The TSF shall run a suite of self-**tests in accordance with FIPS PUB 140-2, Level 4 (as identified in Table 5.3)** <u>during initial start-up</u> **(on power on)**, <u>at the request of the</u> **cryptographic administrator (on demand)**, **under various conditions,** <u>and periodically</u> **(at least once a day)** to demonstrate the correct operation of the ~~TSF~~ **following …**

**ii** A deletion of CC text was performed in FPT_TST.1.2(2). Rationale: The word "users" was deleted to replace it with the role of " cryptographic administrator". "Only authorized cryptographic administrators should be given the capability to verify the integrity of cryptographically related TSF data.

> FPT_TST.1.2(1) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of **TSF data related to the cryptography by using TSF-provided cryptographic functions.**.

**iii** A deletion of CC text was performed in FPT_TST.1.3(1). Rationale: The word "users" was deleted to replace it with the role of " cryptographic administrator". Only authorized cryptographic administrators should be given the capability to verify the integrity of cryptographically related TSF executable code.

> FPT_TST.1.3(1) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of stored **cryptographically related** TSF executable code.

**iv** A deletion of CC text was performed in FPT_TST.1.1(2). Rationale: The words "the TSF" was deleted to allow for the demonstration of the correct operation of each key generation component. The word "perform" replaced "run a suite of" for clarity and better flow of the requirement.

> FPT_TST.1.1(2) **Refinement:** The TSF shall ~~run a suite of~~ **perform** self-tests **immediately after generation of a key** to demonstrate the correct operation of ~~the TSF~~ **each key generation component. If any of these tests fails, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140 for failing a self-test, and this event will be audited.**

**v** A deletion of CC text was performed in FPT_TST.1.2(2). Rationale: The word "users" was deleted to replace it with the role of "cryptographic administrator".

> FPT_TST.1.2(2) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of TSF data **related to the key generation**.

**vi** A deletion of CC text was performed in FPT_TST.1.3(2). Rationale: The word "users" was deleted to replace it with the role of "cryptographic administrator".

> FPT_TST.1.3(2) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of stored TSF executable code **related to the key generation**.