U.S. Government

Virtual Private Network (VPN) Boundary Gateway Protection Profile

For

Medium Robustness Environments

Information Assurance Directorate

Version 1.2

30 January 2009

PREFACE

Protection Profile Title:

U.S. Government Protection Profile Virtual Priviate Network (VPN) for Medium Robustness Environments

Criteria Version:

This Protection Profile "*US Government Protection Profile Virtual Priviate Network (VPN) 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.

RECORD OF RELEASE

(Back to TOC)

Release #	Date	Area Affected	Comment
Release 1.0	February 23, 2006	Complete Document	Release of the NIAP evaluated PP
Release 1.1	July 25, 2007	Assurance requirements Functional requirements NIAP Interps Rationale	Updated to CC version 3.1, updated FCS SFR with current standards.
Release 1.2	January 30, 2009	Crypto	Modified crypto requirements based on comments

Protection Profile Title:

U.S. Government Virtual Private Network (VPN) Protection Profile for Medium Robustness Environments.

Criteria Version:

This Protection Profile (PP) was updated using Version 3.1 of the Common Criteria (CC) [1] and applying the NIAP interpretations that have been approved by TTAP/CCEVS Management as of July 10, 2002.

Table of Contents

PREFACE .... **I**

RECORD OF RELEASE .... **II**

1 INTRODUCTION TO THE PROTECTION PROFILE **1**

1.1 PP IDENTIFICATION 1
1.2 OVERVIEW OF THE PROTECTION PROFILE 1
1.3 CONVENTIONS 2
1.4 GLOSSARY OF TERMS 3
1.5 DOCUMENT ORGANIZATION 3

2 TOE DESCRIPTION **5**

2.1 PRODUCT TYPE 5
2.2 TOE DEFINITION 5
2.3 GENERAL TOE FUNCTIONALITY 5
2.4 TOE OPERATIONAL ENVIRONMENT 9

3 SECURITY ENVIRONMENT **9**

3.1 THREATS 10
3.2 ORGANIZATIONAL SECURITY POLICIES 13
3.3 ASSUMPTIONS 14

4 SECURITY OBJECTIVES **15**

4.1 TOE SECURITY OBJECTIVES 15
4.2 ENVIRONMENT SECURITY OBJECTIVES 17

5 IT SECURITY REQUIREMENTS **19**

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 19
5.2 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT 71
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 73

6 RATIONALE **91**

6.1 RATIONALE FOR TOE SECURITY OBJECTIVES 91
6.2 RATIONALE FOR THE SECURITY OBJECTIVES AND SECURITY FUNCTIONAL REQUIREMENTS FOR THE ENVIRONMENT 110
6.3 RATIONALE FOR TOE SECURITY REQUIREMENTS 110
6.4 RATIONALE FOR ASSURANCE REQUIREMENTS 138
6.5 RATIONALE FOR SATISFYING ALL DEPENDENCIES 138
6.6 RATIONALE FOR EXTENDED REQUIREMENTS 140

7 APPENDICES **143**

List of Tables

Table 1 Medium Robustness Applicable Threats 11

Table 2 Organizational Security Policies 13

Table 3 Medium Robustness Applicable Assumptions 14

Table 4 Medium Robustness Security Objectives 15

Table 5 Environmental Security Objectives 17

Table 6 Security Functional Requirements 19

Table 7 Auditable Events Table 24

Table 8 Assurance Requirements 73

Table 9 Rationale for TOE Security Objectives 91

Table 10 Requirements vs. Objectives Mapping 110

Table 11 Requirement Dependencies 138

Table 12 Rationale for Extended Requirements 140

1 INTRODUCTION TO THE PROTECTION PROFILE

1.1 PP IDENTIFICATION

1 Title: U. S. Government Virtual Private Network (VPN) Boundary Gateway Protection Profile (PP) for Medium Robustness Environments

2 Sponsor: National Security Agency (NSA)

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

4 Registration: <to be provided upon registration>

5 PP Version: Version 1.1 dated 25 July 2007

6 Keywords: Virtual Private Network, VPN, protection profile, Gateway Boundary, encryption, decryption, IPSEC ESP, IKE

1.2 OVERVIEW OF THE PROTECTION PROFILE

7 The US Government Virtual Private Network (VPN) Boundary Gateway Protection Profile (PP) for Medium Robustness Environments was generated under the Enclave Boundary Security Technologies and Solutions (EBST&S) Support Program, sponsored by the National Security Agency (NSA). This PP is intended to be used as follows:

- For product vendors and security product 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. By matching the PP with available STs, security gaps may be identified and products or procedures may be configured to bridge these gaps.

8 This PP specifies the minimum-security requirements for VPN devices (hereafter referred to as the Target of Evaluation (TOE)) used by the Department of Defense (DoD) in Medium Robustness Environments. 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.

9 The TOE may consist of one or more devices that act as part of an organization’s overall security defense by encrypting traffic flowing between enclaves that are geographically separated. If the security policy specifies encryption, the TOE automatically encrypts all outgoing traffic from the enclave when it is destined for another enclave having the same security policy. If the security policy does not specify encryption, all outgoing traffic will be sent unencrypted. The TOE

decrypts incoming traffic to the enclave when that traffic has been encrypted at the originating enclave.

10 The TOE supports identification and authentication for the administrative roles (I&A). The TOE shall generate audit data of security relevant events and will meet the assurance requirements of Evaluation Assurance Level (EAL) 4 augmented.

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

12 This PP defines:

- 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.

1.3 CONVENTIONS

13 Except for replacing United Kingdom spelling with American spelling, the notation, formatting, and conventions used in this PP are consistent with version 3.1 of the CC. Selected presentation choices are discussed here to aid the PP reader.

14 The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph C4 of Part 1 of the CC. Each of these operations is used in this PP.

15 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**.

16 The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted by *italicized text*, selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are not italicized.

17 The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing the value in square brackets, [Assignment_value], assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:].

18 The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

19 As this PP was sponsored, in part by NSA, National Information Assurance Partnership (NIAP) interpretations are used and are presented with the NIAP interpretation number as part of the requirement identifier (e.g., **FAU_GEN.1-NIAP-0407** for Audit data generation).

20 The CC paradigm also allows protection profile and security target authors to create their own requirements as defined in paragraph C5 of Part 1 of the CC. Such requirements are termed 'extended requirements' and are permitted if the CC does not offer suitable requirements to meet the authors' needs. Extended requirements must be identified and are required to use the CC class/family/component model in articulating the requirements. In this PP, extended requirements will be indicated with the "(EXT)" following the component name (e.g., FCS_CKM_(EXT).2).

21 Application Notes are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

1.4 Glossary of Terms

22 See appendix B for the Glossary.

1.5 Document Organization

23 Section 1, Protection Profile Introduction, provides document management and overview information necessary to identify the PP along with references to other related PP's.

24 Section 2, Target of Evaluation (TOE) Description, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

25 Section 3, TOE Security Environment (TSE), describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

26 Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

27 Section 5, IT Security Requirements, defines the security functional and assurance requirements that must be satisfied by the TOE and the Non-IT environment.

28 Section 6, Rationale, provides rationale to demonstrate that the security objectives satisfy the threats and policies. This section also explains how the set of requirements are complete relative to the security objectives and presents a set of arguments that address dependency analysis and use of the extended requirement.

29 Appendix A, References, provides background material for further investigation by users of the PP.

30 Appendix B, Glossary, provides a listing of definitions of terms.

31 Appendix C, Acronyms, provides a listing of acronyms used throughout the document.

32 Appendix D, Characterization of Robustness, contains a discussion characterizing the level of robustness TOEs compliant with the PP can achieve. The PPRB created a discussion that provides a definition of factors for TOE environments as well as an explanation of how a given level of robustness is categorized.

33 Appendix E, Refinements, identifies the refinements that were made to CC requirements where text is deleted from a requirement.

34 Appendix F, Statistical Number Generator Tests, describes the statistical tests that must be performed to the random number generators.

2 TOE DESCRIPTION

35 This Protection Profile specifies the minimum security requirements to satisfy Medium Robustness Environments for a TOE that is a VPN.

2.1 PRODUCT TYPE

36 A VPN boundary gateway is a component that performs encryption and decryption of IP packets as they cross the boundary between a private network and a public network. IP packets crossing from the private network to the public network will be encrypted if their destination is to another private network supporting the same VPN policy as the source network. Encryption of all packets between the two networks assures that the data communicated between the two networks is kept private, eventhough it traverses a public network.

2.2 TOE DEFINITION

37 A VPN provides the ability to use a public network, such as the Internet, as if it were a secure, private network. A VPN is created through the use of devices that can establish secure communication channels over a common, untrusted (or less trusted) communications infrastructure, protecting data in-transit between two communicating entities.[1] The secure communications channels are established using security mechanisms such as encryption, digital signatures, identification and authentication, and access controls. Such secure communications channels may be established over Local Area Networks (LANs), Campus Area Networks (CANs), Metropolitan Area Networks (MANs), privately owned Wide Area Networks (WANs), or public WANs (e.g., the Internet).

2.3 GENERAL TOE FUNCTIONALITY

38 A VPN System that is complaint with the VPN PP provides the following security functions in its evaluated configuration:

- Identification and Authentication –The TOEs will exchange identities and will perform two types of authentication: device-level authentication of the remote device (peer TOEs, remote VPN gateways or VPN clients) and user authentication of the Authorized Administrator. Device-level authentication enables a TOE to construct a secure channel with a trusted peer. The secure channel should be established only after each device authenticates itself. Device-level authentication is performed using authentication techniques specified in RFC 2409. The TOE will assure that the trust establishment is

[1] This is often referred to as a *Secure VPN Tunnel*.

mutual. In other words, peers will mutually authenticate themselves to each other before establishing the secure channel.

- “Administrators” refers to the roles assigned to the individuals responsible for the installation, configuration, and maintenance of the TOE. The TOE requires three separate administrative roles: Cryptographic Administrator, Audit Administrator and Security Administrator. The Cryptographic Administrator is responsible for the configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the TOE. The Audit Administrator is responsible for the regular review of the TOE’s audit data. The Security Administrator is responsible for all other administrative tasks (e.g., creating the TOE security policy) not addressed by the other two administrative roles. It is important to note that while this PP requires the three administrative roles outlined above, it provides the ST author the option of including additional administrative roles as well.

- Audit – Section 5.1.1 “Security Audit (FAU)” describes the TOE’s generation of auditable events, audit records, alarms and audit management. Table 6 in the FAU_GEN.1-NIAP-0407 requirement lists the minimum set of auditable events that must be available to the Security Administrator for configuration on the TOE. Each auditable event must generate an audit record. Table 7 also provides a minimum list of attributes that must be included in each audit record. The ST author may include additional auditable events and audit record attributes. If the ST author includes any additional functional requirements not specified by this PP, they must consider any security relevant events associated with those requirements and include them in the TOE’s list of auditable events and records. 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 administrator console (all administrative roles included). Additionally, the Security Administrator can configure the TOE to generate an audible alarm to indicate a potential security violation. If an administrator console is not active, the TOE stores the message for display when the console becomes active (e.g. when the administrator establishes a remote session to the TOE). The message must contain the potential security violation and all audit records associated with the potential security violation. The message will be displayed at the various consoles until administrator acknowledgement of the message has occurred. As mentioned in the “Administrative” section above, the Audit Administrator’s role is restricted to viewing the contents of the audit records and the deletion of the audit trail. The TOE does provide the Audit Administrator with a sorting and searching capability to improve audit analysis. The Security Administrator configures auditable events, backs-up and deletes audit data, and manages audit data storage. The TOE provides the Security Administrator with a configurable audit trail threshold to track the storage capacity of the audit trail. As soon as the threshold is met, the TOE generates an alarm and displays a message in the same fashion as described above, including the option of the audible alarm. In addition to displaying the message, the Security Administrator may configure the TOE to prevent all auditable events except for those performed by the Security and Audit Administrators or overwrite the oldest audit records in the audit trail.

Audit events include modifications to the group of individuals associated with the Authorized Administrator roles; use of the identification and authentication mechanisms (including any attempted reuse of authentication data); changes made to the TOE's security policy rules, mechanisms and data; actions taken due to imminent security violations; decisions made by the TOE to enforce security policy rules; changes to the TOE's date and time; and the use of other security functions. The decision to record auditable events will be made in accordance with organizational security policy and implemented by the Authorized Administrator. If the audit trail becomes full then the only auditable events that are recorded are those performed by the Authorized Administrator. Audit trail data is stamped with a dependable date and time when recorded.

- Trusted Channel/ Trusted Path- The TOE is required to provide two types of encrypted communications: trusted channel and trusted path. Trusted channel refers to the encrypted connection between the TOE and a non-human external source. An encrypted connection between the TOE and authorized Information Technology (IT) entities (e.g., NTP server, certificate authority) is an example of trusted channel encryption. Trusted path refers to the encrypted connection used to authenticate an external human user with the TOE. Remote administrators establishing an encrypted link to authenticate to the TOE are examples of trusted path encryption. The remote administrator's communication must remain encrypted throughout the remote session.

- Encryption – As mentioned in the paragraph above, the TOE must establish encrypted communications (acting as the initiator or responder) with authorized remote users and external IT entities. Section 5.1.2 "Cryptographic Support" defines the minimum set of cryptographic attributes required by the TOE. The TOE's cryptographic module(s) must be FIPS PUB 140-2 validated and must meet, as a minimum, the security requirements of "Security Level 1". The ST author may implement the cryptographic module(s) in hardware, software, or a combination of both. The TOE must generate and distribute symmetric and asymmetric keys. The ST author is provided several implementation selections for key generation and may distribute keys manually, electronically, or both. The TOE must perform data encryption/decryption using the Advanced Encryption Standard (AES) algorithm with a minimum key size of 128 bits. Additional requirements for key destruction, digital signature generation/verification, random number generation and cryptographic hashing are provided in section 5.1.2.

  The TOE shall implement VPN mechanisms using cryptography, key management, access control, authentication, and data integrity. TOEs meeting this PP will implement and conform to the Internet Engineering Task Force (IETF) Internet Protocol Security (IPSEC) Encapsulating Security Payload (ESP) protocol as specified in RFC 2406. The TOE can also use any of the ESP transforms described in RFC 4869 to provide confidentiality and integrity for user traffic. All VPN traffic between peer TOEs shall use tunnel mode, support for transport mode is optional. TOE encryption mechanisms will conform to IETF *ESP CBC-Mode Cipher Algorithms* as specified in RFC 2451. The TOE shall, at a minimum, implement the Rijndael algorithm as specified in the Advanced Encryption Standard (AES), FIPS PUB 197. TOE data integrity mechanisms will conform to IETF *Use of HMAC-SHA-1-96 within ESP and AH* as specified in RFC 2404. The TOE shall utilize cryptographic modules that are compliant with FIPS PUB 140-2. The TOE shall perform key management and key exchange using the IETF specified Internet Key Exchange (IKE) (RFC 2409) which shall be FIPS PUB 140-2 compliant.

- Information Flow Control – The TOE supports two information flow control policies: VPN and unauthenticated TOE services. The TOE's VPN SFP is instantiated by a device at each enclave boundary. The TOE is a VPN functional component that may either be hosted on a firewall or router, or may be a dedicated VPN gateway device. If the TOE is a firewall or router with VPN capability, the entire device, including all software and hardware that can affect the security functions and assurances of the VPN must meet the assurance requirements of this protection profile. Each TOE authenticates itself to the remote device (peer TOE, remote VPN gateway or VPN client), agrees upon cryptographic keys and algorithms, securely generates and distributes session keys as necessary, and encrypts network traffic in accordance with the TOE security policy. The TOE will enforce the same security policy between communicating peers.

  The TOE will enforce a security policy as follows:

  - for outbound traffic associated with a peer TOE, a remote VPN gateway or a VPN client, the local TOE will create or use an existing secure channel between the remote device if there exists an information flow control rule specifying that communication between the source and destination IP addresses must be encrypted;

  - for outbound traffic not associated with a peer TOE, remote VPN gateway or VPN client, the local TOE will not invoke the security mechanisms and a secure channel will not be established;

  - for inbound traffic associated with a peer TOE, remote VPN gateway or VPN client, the local TOE will create or use an existing secure channel between the devices if there exists an information flow control rule specifying that communication between the source and destination IP addresses must be encrypted; and

  - for inbound network traffic not associated with a peer TOE, remote VPN gateway or VPN client, the local TOE will not invoke the security mechanisms and a secure channel will not be established.

  The unauthenticated TOE services information flow control policy supported by the TOE provides the rules that apply to the unauthenticated use of any services provided by the TOE. ICMP is the only service that is required to be provided by the TOE, and the security attributes associated with this protocol allow the Security Administrator to specify what degree the ICMP traffic is mediated (i.e., the ICMP message type and code).

2.4 TOE Operational Environment

39 The operational environment for the TOE is at the boundary between a private network and a less-trusted network (e.g., the Internet). While the VPN gateway is a part of the private network, and its primary function is to protect data communication between private networks, it is exposed to threats from the less-trusted network.

3 Security Environment

40 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 while highly sophisticated threat agents will not be motivated to use great expertise or extensive resources in an environment where medium robustness is suitable, the wide spread availability of exploits and hacking tools available on the Internet provide less sophisticated threat agents with expertise (and indirectly resources) that they otherwise might not have access to.

41 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).

42 The remainder of this section addresses the following:

- 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.

43 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 prior to the fielding of the TOE.

3.1 THREATS

3.1.1 Threat Agent Characterization

44 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).

45 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.

46 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.

47 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*.

48 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.

49 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.

50 The important general points we can make are:

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

51 The following threats are addressed by the TOE and should be read in conjunction with the threat rationale section. There are other threats that the TOE does not address (e.g., malicious developer inserting a backdoor into the TOE) and it is up to a site to determine how these types of threats apply to its environment.

Table 1 Medium Robustness Applicable Threats

Threat Name	Threat Definition
T.ADDRESS_MASQUERADE	A user on one interface may masquerade as a user on another interface to circumvent the TOE policy.
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.

Threat Name	Threat Definition
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.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.FLAWED_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.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 may masquerade as an authorized user or an authorized IT entity to gain 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 undiscovered thereby causing potential security vulnerabilities.
T.REPLAY	A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes (captured as it was transmitted during the course of legitimate use).
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 TOE system resources (e.g., connection state tables) via a resource exhaustion denial of service attack.
T.SPOOFING	An entity may mis-represent itself as the TOE to obtain authentication data.

Threat Name	Threat Definition
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain access to services (either on the TOE or by sending data through the TOE) for which they are not authorized according to the TOE security policy.
T.UNAUTHORIZED_PEER	An unauthorized IT entity may attempt to establish a security association with the TOE.
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, design flaws, or improper configurations may cause the security state of the TOE to be unknown.

3.2 ORGANIZATIONAL SECURITY POLICIES

52 An Organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This section identifies the Organizational security policies applicable to the VPN PP.

Table 2 Organizational Security Policies

Policy Name	Policy
P.ACCESS_BANNER	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.
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.CRYPTOGRAPHIC_FUNCTIONS	The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.

Policy Name	Policy
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.INTEGRITY	The TOE shall support the IETF *Internet Protocol Security Encapsulating Security Payload* (IPSEC ESP) as specified in RFC 2406. Sensitive information transmitted to a peer TOE shall apply integrity mechanisms as specified in *Use of HMAC-SHA-1-96 within ESP and AH* (RFC 2404).
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.

3.3 ASSUMPTIONS

53 This section contains assumptions regarding the security environment and the intended usage of the TOE.

Table 3 Medium Robustness Applicable Assumptions

Name	Definition
A.NO_GENERAL_PURPOSE	There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.NO_TOE_BYPASS	Information cannot flow between external and internal networks located in different enclaves without passing through the TOE.

4 SECURITY OBJECTIVES

54 This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.1 TOE SECURITY OBJECTIVES

55 The following are the TOE security objectives:

Table 4 Medium Robustness Security Objectives

Objective Name	Definition
O.ADMIN_ROLE	The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.
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 in its operational environment.
O.CRYPTOGRAPHIC_ FUNCTIONS	The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.
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.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.

Objective Name	Definition
O.INTEGRITY	The TOE must be able to protect the integrity of data transmitted to a peer TOE via encryption and provide IPSec authentication for such data. Upon receipt of data from a peer TOE, the TOE must be able to decrypt the data and verify that the received data accurately represents the data that was originally transmitted.
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 mediate the flow of information between sets of TOE network interfaces or between a network interface and the TOE itself in accordance with its security policy.
O.PEER_AUTHENTICTION	The TOE will authenticate each peer TOE that attempts to establish a security association with the TOE.
O.REPLAY_DETECTION	The TOE will provide a means to detect and reject the replay of TSF data and security attributes.
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 connection-oriented resources provided by the TOE (e.g., entries in a connection state table; Transmission Control Protocol (TCP) connections to 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

Objective Name	Definition
	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 ENVIRONMENT SECURITY OBJECTIVES

56 The TOE's operating environment must satisfy the following objectives. These objectives do not levy any IT requirements but are satisfied by procedural or administrative measures.

Table 5 Environmental Security Objectives

Environmental Objective Name	Environmental Objective Definition
OE.CRYPTANALYTIC	Cryptographic methods used in the IT environment shall be interoperable with the TOE, should be FIPS 140-2 validated and should be resistant to cryptanalytic attacks (i.e., will be of adequate strength to protect unclassified Mission Support, Administrative, or Mission Critical data).
OE.NO_GENERAL_PURPOSE	The Administrator ensures there are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

OE.NO_TOE_BYPASS	Information cannot flow between external and internal networks located in different enclaves without passing through the TOE.
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.

5 IT SECURITY REQUIREMENTS

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

57 This section defines the functional requirements for the TOE. Functional requirements in this PP were drawn from Part 2 of the CC or were based on Part 2 of the CC. These requirements are relevant to supporting the secure operation of the TOE. The functional security requirements for the PP consist of the following components, summarised in Table 6.

Table 6 Security Functional Requirements

Functional Components (from CC Part 2)	
FAU_ARP.1	Security alarms
FAU_ARP_ACK_(EXT).1	Security alarm acknowledgement
FAU_GEN.1-NIAP-0407	Audit data generation
FAU_GEN.2-NIAP-0410	User identity association
FAU_SAA.1-NIAP-0407	Potential violation analysis
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable audit review
FAU_SEL.1-NIAP-0407	Selective Audit
FAU_STG.1-NIAP-0429	Protected audit trail storage
FAU_STG.3	Action in case of possible audit data loss
FAU_STG.NIAP-0414-1-NIAP-0429	Site-Configurable Prevention of Audit Loss
FCS_BCM_(EXT).1	Baseline Cryptographic Module
FCS_CKM.1(1)	Cryptographic key generation (for symmetric keys using RNG)
FCS_CKM.1(2)	Cryptographic key generation (for asymmetric keys)

Functional Components (from CC Part 2)	
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 data encryption/decryption)
FCS_COP.1(2)	Cryptographic operation (for cryptographic signature)
FCS_COP.1(3)	Cryptographic operation (for cryptographic hashing)
FCS_COP.1(4)	Cryptographic operation (for cryptographic key agreement)
FCS_COP_(EXT).1	Random Number Generation
FCS_IKE_(EXT).1	Internet Key Exchange
FDP_IFC.1(1)	Subset information flow control (VPN policy)
FDP_IFC.1(2)	Subset information flow control (unauthenticated TOE services policy)
FDP_IFF.1(1)	Simple security attributes (VPN policy)
FDP_IFF.1(2)	Simple security attributes (unauthenticated TOE services policy)
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication (for TOE-provided services)
FIA_UAU.2	User authentication before any action
FIA_UAU_(EXT).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 behavior (TSF non-cryptographic self-test)
FMT_MOF.1(2)	Management of security functions behavior (cryptographic self-test)
FMT_MOF.1(3)	Management of security functions behavior (audit and alarms)
FMT_MOF.1(4)	Management of security functions behavior (audit and alarms)
FMT_MOF.1(5)	Management of security functions behavior (audit and alarms)
FMT_MOF.1(6)	Management of security functions behavior (available TOE-services

Functional Components (from CC Part 2)	
	for unauthenticated users)
FMT_MOF.1(7)	Management of security functions behavior (quota mechanism)
FMT_MSA.1	Management of security attributes
FMT_MSA.3 (1)	Static attribute initialization (ruleset)
FMT_MSA.3 (2)	Static attribute initialization (services)
FMT_MTD.1(1)	Management of TSF data (non-cryptographic, non-time TSF data)
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 (VPN Policy Ruleset)
FMT_MTD.2(1)	Management of limits on TSF data (transport-layer quotas)
FMT_MTD.2(2)	Management of limits on TSF data (controlled connection-oriented quotas)
FMT_REV.1	Revocation
FMT_SMR.2	Restrictions on security roles
FPT_RCV.1	Manual Recovery
FPT_RPL.1	Replay detection
FPT_STM.1	Reliable time stamps
FPT_TST.1(1)	Cryptographic self-test
FPT_TST.1(2)	Key Generation self test
FPT_TST_(EXT).1	TSF testing
FRU_RSA.1(1)	Maximum quotas (transport-layer quotas)
FRU_RSA.1(2)	Maximum quotas (controlled connection-oriented quotas)
FTA_SSL.1	TSF-initiated session locking
FTA_SSL.2	User-initiated locking
FTA_SSL.3	TSF-initiated termination
FTA_TAB.1	Default TOE access banners
FTA_TSE.1	TOE session establishment
FTP_ITC.1(1)	Inter-TSF trusted channel (Prevention of Disclosure)
FTP_ITC.1(2)	Inter-TSF trusted channel (Detection of Modification)
FTP_TRP.1(1)	Trusted path (Prevention of Disclosure)

Functional Components (from CC Part 2)	
FTP_TRP.1(2)	Trusted path (Detection of Modification)

5.1.1 Security audit (FAU)

5.1.1.1 FAU_ARP.1 Security alarms

FAU_ARP.1.1 – **Refinement:** The TSF shall [immediately display an alarm 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:

a) local console,

b) remote administrator sessions that exist, and;

c) remote administrator sessions that are initiated before the alarm has been acknowledged, and;

d) at the option of the Security Administrator, generate an audible alarm, and;

e) [selection: [assignment: other methods determined by the ST author], "no other methods"]]

upon detection of a potential security violation.

58 *Application Note: The TSF provides a message to the local console regardless of whether an administrator is logged in. The message is displayed at the remote console if an administrator is already logged in, or when an administrator logs in if the alarm message has not 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 administrators. In addition, the TOE provides an audible alarm that can be configured to sound an alarm if desired by the Security Administrator. It is acceptable for the ST author to fill the open assignment with none, if no other methods (e.g., pager, e-mail) are included in the TOE.*

5.1.1.2 FAU_ARP_ACK_(EXT).1 Security alarm acknowledgement

FAU_ARP_ACK_(EXT).1.1 – The TSF shall display the alarm message identifying the potential security violation and make accessible the audit record contents associated with the auditable event(s) until it has been acknowledged. An audible alarm will sound until acknowledged by an administrator.

FAU_ARP_ACK_(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:

a) local console, and

b) remote administrator sessions that received the alarm.

59 *Application Note: 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 they have 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). If the Security Administrator configures the TOE to generate an audible alarm, the alarm will sound until an administrator acknowledges the alarm. Acknowledging the message and audible alarm could be a single event, or different events.*

60 *FAU_ARP_ACK_(EXT).1.2 ensures that each administrator that received the alarm message also receives the acknowledgement message, which includes some form of reference to the alarm message, who acknowledged the message and when.*

5.1.1.3 FAU_GEN.1-NIAP-0407 Audit data generation

FAU_GEN.1.1-NIAP-0407 – **Refinement:** The TSF shall be able to generate an audit record of the following auditable events:

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

b) All auditable events [listed in Table 7];

c) [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].

61 *Application Note: For the first assignment in the selection, 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.*

62 *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. It is acceptable for the ST author to choose "no additional events", if the ST author has not included additional requirements, or has included additional requirements that do not have a basic level (or commensurate level) of audit associated with them.

FAU_GEN.1.2-NIAP-0407 - The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, 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 7 below].

63 *Application Note: In column 3 of the table 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. For example, in FDP_IFF, packets may be allowed to flow that do not have a transport layer component (e.g., an ICMP Echo request). For those packets, there is nothing to record with respect to the transport layer abstractions.*

Table 7 Auditable Events Table

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

Requirement	Auditable Events	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	The identity of the Security Administrator performing the function
FAU_STG.NIAP-0414-1-NIAP-0429	Actions taken due to the audit storage failure	The identity of the Security Administrator performing the function
FCS_BCM_(EXT).1	None	
FCS_CKM.1(1)	Generation and loading of key. Failure of the activity	
FCS_CKM.1(2)	Generation and loading of key pair for digital signatures. Failure of the activity	
FCS_CKM.2	None	
FCS_CKM.4	None	
FCS_CKM.(EXT).1	None	
FCS_CKM.(EXT).2	None	
FCS_COP.1(1)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP.1(2)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information

Requirement	Auditable Events	Audit Record Contents
FCS_COP.1(3)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP.1(4)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP_(EXT).1	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_IKE_(EXT).1	Generation and loading of key pair for digital signatures. Changes to the pre-shared key used for authentication All modifications to the key lifetimes. Failure of the authentication in Phase 1. Failure to negotiate a security association in Phase 2.	If failure occurs, record an English description for the failure.
FDP_IFC.1(1)	None	
FDP_IFC.1(2)	None	

Requirement	Auditable Events	Audit Record Contents
FDP_IFF.1(1)	Decisions to permit or deny information flows. Operation applied to each information flow permitted.	Presumed identity of source subject Identity of destination subject Transport layer protocol, if applicable Source subject service identifier, if applicable Destination subject service identifier, if applicable Identity of the interface on which the TOE received the packet For denied information flows, the reason for denial
FDP_IFF.1(2)	Decisions to permit or deny information flows	Presumed identity of source subject Identity of destination subject Transport layer protocol, if applicable Source subject service identifier, if applicable Destination subject service identifier, if applicable Identity of the interface on which the TOE received the packet For denied information flows, the reason for denial
FDP_RIP.2	None	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts The actions (e.g. disabling of an account) taken	Identity of the unsuccessfully authenticated user

Requirement	Auditable Events	Audit Record Contents
	The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Identity of the unsuccessfully authenticated user and the identity of the administrator performing the function.
FIA_ATD.1	None	
FIA_UAU.1	None	
FIA_UAU.2	Successful and unsuccessful use of authentication mechanisms	Claimed identity of the user using the authentication mechanism
FIA_UAU_(EXT).5	All use of the local authentication mechanism	Claimed identity of the user attempting to authenticate
FIA_UID.2	All use of the user identification mechanism used for authorized users (that is, those that authenticate to the TOE)	Claimed identity of the user using the identification mechanism
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(1)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(2)	Enabling or disabling of the key-generation self-tests	The identity of the administrator performing the function
FMT_MOF.1(3)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(4)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(5)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(6)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(7)	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MSA.1	All manipulation of the security attributes	The identity of the administrator performing the function

Requirement	Auditable Events	Audit Record Contents
FMT_MSA.3 (1)	None	
FMT_MSA.3 (2)	None	
FMT_MTD.1(1)	All modifications of the values of TSF data by the administrator	The identity of the administrator performing the function
FMT_MTD.1(2)	All modifications of the values of cryptographic security data by the cryptographic administrator	The identity of the administrator performing the function
FMT_MTD.1(3)	All modifications to the time and date used to form the time stamps by the administrator	The identity of the administrator performing the function
FMT_MTD.1(4)	All modifications to the information flow policy ruleset by the Security Administrator	The identity of the security administrator performing the function
FMT_MTD.2(1)	All modifications of the limits Actions taken when the quota is exceed (include the fact that the quota was exceeded)	The identity of the administrator performing the function
FMT_MTD.2(2)	All modifications of the limits Actions taken when the quota is exceed (include the fact that the quota was exceeded)	The identity of the administrator performing the function
FMT_REV.1	All attempts to revoke security attributes	List of security attributes that were attempted to be revoked 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 The identity of the administrator performing the function

Requirement	Auditable Events	Audit Record Contents
FPT_RCV.1	The fact that a failure or service discontinuity occurred Resumption of the regular operation	Type of failure or service discontinuity
FPT_RPL.1 (including replay of authentication data notification from the authentication server)	Notification that a replay event occurred	Identity of the user that was the subject of the reply attack
FPT_STM.1	Changes to the time	The identity of the adminstrator if the change was performed by an administrator or the network identifier of the NTP server if the change was performed from an NTP server.
FPT_TST.1(1)	Execution of this set of Crypto TSF self tests	The identity of the administrator performing the test, if initiated by an administrator
FPT_TST.1(2)	Execution of this set of Key Generation self tests	The identity of the administrator performing the test, if initiated by an administrator
FPT_TST_(EXT).1	Execution of this set of TSF self tests	The identity of the administrator performing the test, if initiated by an administrator
FRU_RSA.1(1)	None	
FRU_RSA.1(2)	None	
FTA_SSL.1	a) Locking of an interactive session by the session locking mechanism. b) Successful unlocking of an interactive session. c) Any attempts at unlocking an interactive session.	The identity of the user associated with the session being locked or unlocked

Requirement	Auditable Events	Audit Record Contents
FTA_SSL.2	a) Locking of an interactive session by the session locking mechanism. b) Successful unlocking of an interactive session. c) Any attempts at unlocking 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	a) Denial of a session establishment due to the session establishment mechanism. b) All attempts at establishment of a user session.	The identity of the user attempting to establish the session For unsuccessful attempts, the reason for denial of the establishment attempt
FTP_ITC.1(1)	a) All attempted uses of the trusted channel functions. b) Identifier of the initiator and target of all trusted channel functions.	Identification of the initiator and target of all trusted channels
FTP_ITC.1(2)	a) All attempted uses of the trusted channel functions. b) Identifier of the initiator and target of all trusted channel functions.	Identification of the initiator and target of all trusted channels
FTP_TRP.1(1)	a) All attempted uses of the trusted path functions. b) Identification of the user associated with all trusted path invocations, if available.	Identification of the claimed user identity

Requirement	Auditable Events	Audit Record Contents
FTP_TRP.1(2)	a) All attempted uses of the trusted path functions. b) Identification of the user associated with all trusted path invocations, if available.	Identification of the claimed user identity

5.1.1.4 FAU_GEN.2-NIAP-0410 User identity association

FAU_GEN.2.1-NIAP-0410 – **Refinement**: The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

64 *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. User in this requirement is the userid for authorized users, and a network identifier for unauthenticated network traffic.*

5.1.1.5 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 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 audited events:

a) [Security Administrator specified number of authentication failures;

b) Security Administrator specified number of Information Flow policy violations by an individual presumed source network identifier (e.g., IP address) within an administrator specified time period;

c) Security Administrator specified number of Information Flow policy violations to an individual destination network identifier within an administrator specified time period;

d) Security Administrator specified number of Information Flow policy violations to an individual destination subject service identifier (e.g., TCP port) within an administrator specified time period;

e) Security Administrator specified Information Flow policy rule, or group of rule violations within an administrator specified time period;

f) Any detected replay of TSF data or security attributes;

g) Any failure of the cryptomodule self-tests (FPT_TST.1(1));

h) Any failure of the other key generation self-tests (FPT_TST.1(2));

i) Any failure of the other TSF self-tests (FPT_TST_(EXT).1);

j) Security Administrator specified number of encryption failures;

k) Security Administrator specified number of decryption failures;

l) Security Administrator specified number of Phase 1 authentication failures when negotiating the Internet Key Exchange protocol;

m) Security Administrator specified number of failures occur during Phase 2 negotiation; and

n) [selection: [assignment: any other rules], "no additional rules"]]

known to indicate a potential security violation;

65 *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. The Security Administrator settable number of authentication failures in (a) is intended to be the same value as specified in FIA_AFL.1.1.*

5.1.1.6 FAU_SAR.1 Audit review

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

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

66 *Application Note: The role Administrator is intended to mean any user acting in an administrative role.*

5.1.1.7 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 Administrators**.

5.1.1.8 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) source subject identity;

c) destination subject identity;

d) ranges of one or more: dates, times, user identities, subject service identifiers, or transport layer protocol;

e) TOE network interfaces; and

f) [selection: [assignment: other criteria determined by the ST Author], no additional criteria]].

67 *Application Note: Audit data should be capable of being searched and sorted on all criteria specified in a – g, 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 Audit Administrator may want all the audit records for a specified source subject identity or range of source subject identities (e.g., IP source address or range of IP source addresses) 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".*

5.1.1.9 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 auditable events from the set of audited events based on the following attributes:

a) *user identity;*

b) *event type;*

c) [network identifier;

d) subject service identifier;

e) success of auditable security events;

f) failure of auditable security events; and

g) [selection: [assignment: list of additional criteria that audit selectivity is based upon], no additional criteria]].

68 *Application Note: "user identity" applies to authenticated users; see application note for FIA_UID.2. "service identifier" is defined in FDP_IFF.1.2(*). "event type" is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.*

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

FAU_STG.1.1– **Refinement**: The TSF shall **restrict the deletion of** stored audit records **in the audit trail to the Audit Administrator**.

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

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

FAU_STG.3.1 - **Refinement**: The TSF shall [immediately alert the administrators by displaying a message at the local console, and at the remote administrative console when an administrative session exists for each of the defined administrative roles, at the option of the Security Administrator generate an audible alarm, [selection: [assignment: other methods], no other methods]] if the audit trail exceeds [a Security Administrator settable percentage of storage capacity].

69 *Application Note: As with FAU_ARP.1, the TSF provides a message to the local console regardless of whether an administrator is logged in. The message is displayed at the remote console if an administrator is already logged in, or when an administrator logs in. This requirement specifies that the message is sent to the first established session for each of the defined roles to ensure someone in the administrator staff is aware of the alert as soon as possible.*

5.1.1.12 FAU_STG.NIAP-0414-1-NIAP-0429 Site-Configurable Prevention of Audit Loss

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

FAU_STG.NIAP-0414-1.2-NIAP-0429 - **Refinement**: The TSF shall **enforce the Security Administrator's selection(s)** if the audit trail is full.

70 *Application Note: The TOE provides the Security Administrator the option of preventing audit data loss by preventing auditable events from occurring. The Security Administrator and Audit Administrator 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.*

5.1.2 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.

5.1.2.1 Explicit: 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.

Explicit: 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.

5.1.2.2 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.

Explicit: 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 persistent secret and private keys when not in use 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.

5.1.2.3 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 any message digest specified in FIPS 180-2 *[selection: one or more of*** SHA-1, SHA235, SHA-384, or SHA-512***]***.

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."

Explicit: 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.

5.1.2.4 FCS_IKE_(EXT).1 Internet Key Exchange

FCS_IKE_(EXT).1.1 - The TSF shall provide cryptographic key establishment techniques in accordance with RFC 2409 as follows(s):

- Phase 1, the establishment of a secure authenticated channel between the TOE and another remote VPN endpoint, shall be performed using one of the following, as configured by the security administrator:
  - Main Mode
  - Aggressive Mode
- Phase 2, negotiation of security services for IPsec, shall be done using Quick Mode, using SHA-1 as the pseudo-random function. Quick Mode shall generate key material that provides perfect forward secrecy. The use of SHA-256 and SHA-384 as the PRF in IKEv1 KDF is also allowed.

FCS_IKE_(EXT).1.2 - The TSF shall require **the x of g^xy** be randomly generated using a FIPS-approved random number generator when computation is being performed. The minimum size of x shall be twice the number of bits of the strength level associated with the negotiated DH group per table 2 of NIST SP 800-57. The nonce sizes are to be between 8 and 256 bytes. Nounces shall be generated in a manner such that the probability that a specific nounce value will be repeated during the life a specific IPsec SA is less than 1 in 2^(bit strength of the negotiated DH group)

FCS_IKE_(EXT).1.3 - When performing authentication using pre-shared keys, the key shall be generated using the FIPS approved random number generator specified in FCS_COP_(EXT).1.1.

FCS_IKE_(EXT).1.4 - The TSF shall compute the value of SKEYID (as defined in RFC 2409), using SHA-1 as the pseudo-random function. The use of SHA-256 and SHA-384 as the PRF in IKEv1 KDF is also allowed. The TSF shall be capable of authentication using the methods for

- Signatures: SKEYID = prf(Ni_b | Nr_b, g^xy)
- Pre-shared keys: SKEYID = prf(pre-shared-key, Ni_b | Nr_b)
- [*selection: Authentication using Public key encryption, computing SKEYID as follows:* SKEYID = prf(prf(Ni_b | Nr_b), CKY-I | CKY-R), [*assignment: other authentication method],"no other authentication methods"*]

71 *Application Note: If public key encryption is the method of choice, the prf algorithm listed in the requirement will be used. If the other option is selected, a different authentication method or a different hash algorithm for generating SKEYID may be specified.*

72 *Refer to RFC 2409 for an explanation of the notation and definitions of the terms.*

FCS_IKE_(EXT).1.5 - The TSF shall compute authenticated keying material as follows:

- SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
- SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
- SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2)
- [*selection*: [*assignment: other methods for computing the authenticated keying material*], *none*]]

73 *Application Note: If the assignment is selected, a different method for computing the authenticated keying material may be used, or a different hash algorithm may be specified.*

FCS_IKE_(EXT).1.6 - To authenticate the Phase 1 exchange, the TSF shall generate HASH_I if it
is the intiator, or HASH_R if it is the responder as follows:

- HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b)
- HASH_R = prf(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b)

74 *Application Note: Refer to RFC 2409 for an explanation of the notation and definitions of the terms.*

FCS_IKE_(EXT).1.7 - The TSF shall be capable of authenticating IKE Phase 1 using the following methods as defined in RFC 2409, as configured by the security administrator:

a) **Authentication with digital signatures**: The TSF shall use [selection: RSA, DSA,[selection: [assignment: other digital signature algorithms], "no other digital signature algorithms"]]

b) when an RSA signature is applied to HASH I or HASH R it must be first PKCS#1 encoded. The TSF shall check the HASH_I and HASH_R values sent against a computed value to detect any changes made to the proposed transform negotiated in phase one. If changes are detected the session shall be terminated and an alarm shall be generated.

c) [selection:[assignment: X.509 certificates Version 3 [selection: other version of X.509 certificates, "no other versions"]] X.509 V3 implementations, if implemented, shall be capable of checking for validity of the certificate path, and at option of SA, check for certificate revocation.

d) Authentication with a pre-shared key: The TSF shall allow authentication using a pre-shared key.

FCS_IKE_(EXT).1.8. - The TSF shall compute the hash values for Quick Mode in the following way

HASH(1) = prf(SKEYID_a, M-ID |[assignment: any ISAKMP payload after HASH(1) header contained in the message)]

HASH(2) = prf(SKEYID_a, M-ID | Ni_b | [assignment: any ISAKMP payload after HASH(2) header contained in the message)]

HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)

75 *Application Note: The following steps will be performed when using the HASH computation:*

- *initiator computes HASH(1) and sends to responder*
- *responder validates computation of HASH(1) and computes HASH(2) and sends HASH(2) to initiator*
- *initiator validates computation of HASH(2) and computes HASH(3) and sends HASH(3) to responder*

76 *KE is only optional when SA elects not to use perfect forward secrecy.*

77 *Verifying that a TFS implementation actually checks HASH(1) , HASH(2), and HASH(3) values sent against a computed value is important in detecting changes that could have been made to proposed transform negotiated in Quick Mode (not as likely as Phase One because Quick Mode is encrypted).*

78 *The ordering of the ISAKMP payloads may differ because Quick Mode only specifies the location of the HASH and SA payload.*

FCS_IKE_(EXT).1.9 - The TSF shall compute new keying material during Quick Mode as follows:

[selection: when using perfect forward secrecy

KEYMAT = prf(SKEYID_d, g(qm)^xy | protocol | SPI | Ni_b | Nr_b),

When perfect forward secrecy is not used
KEYMAT = prf(SKEYID_d | protocol | SPI | Ni_b | Nr_b)]

FCS_IKE_(EXT).1.10 The TSF shall at a minimum, support the following ID types: ID_IPV4_ADDR, ID_IPV6_ADDR, ID_FQDN, ID_USER_FQDN, [selection:ID_IPV4_ADDR_SUBNET, ID_IPV6_ADDR_SUBNET, ID_IPV4_ADDR_RANGE, ID_IPV6_ADDR_RANGE, ID_DER_ASN1_DN, ID_DER_ASN1_GN, ID_KEY_ID,“no additional ID types”].

79 *Application Note: It should be noted that the Internet Protocol Version 6(Ipv6) Interim Transistion Guidance memorandum, September 29, 2003, provides support to begin to procure/acquire Ipv6 capable GIG assests on October 1, 2003 and a goal for complete transition to Ipv6 at FY2008.*

5.1.3 User data protection (FDP)

5.1.3.1 FDP_IFC.1(1) Subset information flow control (VPN policy)

FDP_IFC.1.1(1) - The TSF shall enforce the [VPN SFP] on

a) [source subject: TOE interface on which information is received;

b) destination subject: TOE interface to which information is destined.

c) information: network packets; and

d) operations:

   i. pass packets without modifying;

   ii. send IPSEC encrypted and authenticated packets to a peer TOE using ESP in tunnel mode as defined in RFC 2406;

   iii. decrypt, verify authentication and pass received packets from a peer TOE in tunnel mode using ESP;

   iv. [assignment:other operations specified in security target]].

80 *Application Note: For this policy, the notion of subject is defined as a physical interface so that we can specify rules that allow the information (packet) to flow from the source subject (the interface the packet comes in on) to the destination subject (the interface the packet is routed to). Note that this policy applies only to flows through the TOE, and not flows that terminate at the TOE itself.*

81 *The term IPSec-authenticated is used throughout this policy to denote the integrity protection applied by the IPSec AH and ESP protocols.*

82 *In a VPN, there are three cases for such flows: the information is allowed to pass through because there is no rule; the information needs to be sent encrypted and/or IPSec-authenticated; and the information is received from a peer TOE encrypted and/or with IPSec authentication information. In the case were the TOE is receiving the information from the peer TOE, the TOE is the destination in the packet, but when the header is stripped (assuming all policy checks succeed) the packet will be decrypted (if necessary) and sent to the destination subject as defined here. See FDP_IFF.1.1 and FDP_IFF.1.2 for more on how this is specified.*

83 *The operations are critical in that they are used to pull in the VPN functionality that makes it distinct from other technologies. A VPN device can allow an information flow without modification of the packet, or can perform a cryptographic operation, such as encryption (ESP), decryption (ESP), generation of IPSec-authentication (ESP or AH), and/or verification of previously-generated IPSec-authentication (ESP or AH). The cryptographic operations are specified by the FCS_COP (how the cryptographic operations work); this component (along with FDP_IFF.1) specifies when these operations are done.*

5.1.3.2 FDP_IFC.1(2) Subset information flow control (unauthenticated TOE services policy)

FDP_IFC.1.1(2) - The TSF shall enforce the [UNAUTHENTICATED TOE SERVICES SFP] on

a) [source subject: TOE interface on which information is received;

b) destination subject: the TOE;

c) information: network packets; and

d) operations: accept or reject network packet].

84 *Application Note: This policy is used to express how the TOE enforces rules concerning network traffic that is destined for the TOE, and the protocols that are allowed as specified in FIA_UAU.1. The intent of this iteration of the requirement is control how the TOE responds to network traffic destined for the TOE, this policy does not have to be enforced in the VPN ruleset (e.g., could be Security Administrator configurable and TOE controlled via another mechanism).*

85 *Note that "operations" refers to the TOE accepting or rejecting the network packet, since the TOE is not technically always providing the "service". In the case of ARP, another machine (e.g., router on the same subnet) is providing an ARP "service" by providing updates to the TOE's routing tables.*

5.1.3.3 FDP_IFF.1(1) Simple security attributes (VPN policy)

FDP_IFF.1.1(1) - The TSF shall enforce the [VPN SFP] based on the following types of subject and information security attributes:

a) [Source subject security attributes:

- set of source subject identifiers; and
- [selection: *[assignment: other subject security attributes determined by the ST Author], none*].

b) Destination subject security attributes:

- Set of destination subject identifiers; and
- [*selection: [assignment: other subject security attributes determined by the ST Author], none*].

86 *Application Note: For the subjects, the administrator knows the set of identifiers that can be associated with the physical VPN interfaces; therefore, they are not "presumed" identifiers. The term "identifiers" was used instead of "addresses" to allow for technologies that are not address-based (e.g., circuit identifiers instead of source and destination addresses).*

87 *The ST author should specify other attributes that are used to identify the source and destination subject sets, based on the technology implemented by the TOE, such as basing the decision on a service identifier as well as a subject identifier (e.g., port number, IP address).*

c) Information security attributes:

- presumed identity of source subject[2];
- identity of destination subject;

FDP_IFF.1.2(1) - **Refinement**: The TSF shall permit an information flow between a **source subject and a destination subject** via a controlled operation if the following rules hold:

- **[**the presumed identity of the source subject is in the set of source subject identifiers;
- the identity of the destination subject is in the set of source destination identifiers;

[2] The TOE can make no claim as to the real identity of any source subject; the TOE can only suppose that such identities are accurate. Therefore, a "presumed identity" is used to identify source subjects. Note, however, that the TOE can ensure that the identity is included in the set that is associated with the interface (see FDP_IFF.1.6(1)).

- the information security attributes match the attributes in an information flow policy rule (contained in the information flow policy ruleset defined by the Security Administrator) according to the following algorithm [assignment: algorithm used by the TOE to match information security attributes to information flow policy rules]; and

- the selected information flow policy rule specifies that the information flow is to be permitted, and what specific operation from FDP_IFC.1(1) is to be applied to that information flow**]**.

88 *Application Note: In a VPN, the administrator specifies information flow policy rules that contain information security attribute values (or wildcards that "stand" for multiple values of the same type; e.g., 127.*.*.* would represent any IP address that begins with "127"), and associate with that rule an action that permits the information flow or disallows the information flow. When a packet arrives at the source interface, the information security attribute values of the packet are compared to each information flow policy rule by some TOE-specified algorithm, and when a match is found the action specified by that rule is taken. Since wildcards would allow the specific attributes in a packet to potentially match more than one rule, the ST author needs to fill in the assignment with the algorithm the TOE uses to find a matching a rule. This could be "first match", "most specific match", or some more elaborate description.*

89 *For the unauthenticated proxies, the security attributes include the SMTP commands that the policy is required to filter and any security attributes associated with additional unauthenticated proxies.*

FDP_IFF.1.3(1) - The TSF shall enforce the [assignment: additional VPN SFP rules]

FDP_IFF.1.4(1) - The TSF shall provide the following [the Security Administrator shall have the capability to view all information flows allowed by the information flow policy ruleset before the ruleset is applied].

90 *Application Note: "before the rule set is applied" means that the administrator is able to view the entire rule set before it is put into use on the TOE. This gives the administrator the opportunity to address any errors or unintended flows.*

FDP_IFF.1.5(1) - The TSF shall explicitly authorize an information flow based on the following rules: [none].

FDP_IFF.1.6(1) - The TSF shall explicitly deny an information flow based on the following rules:

a) [The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE is not included in the set of source identifiers for the source subject;

91 *Application Note: The intent of this requirement is to ensure that a user cannot send packets originating on one TOE interface claiming to originate on another TOE interface.*

b) The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE specifies a broadcast identity;

92 *Application Note: A broadcast identity is one that specifies more than one host address on a network. It is understood that the TOE can only know the sub-netting configuration of networks directly connected to the TOE's interfaces and therefore can only be aware of broadcast addresses on those networks.*

c) The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE specifies a loopback identifier;

d) The TOE shall reject requests in which the information received by the TOE contains the route (set of host network identifiers) by which information shall flow from the source subject to the destination subject.)].

5.1.3.4 FDP_IFF.1(2) Simple security attributes (unauthenticated TOE services policy)

FDP_IFF.1.1(2) - The TSF shall enforce the [UNAUTHENTICATED TOE SERVICES SFP] based on the following types of subject and information security attributes:

a) [Source subject security attributes:

- set of source subject identifiers; and
- [*selection: [assignment: other subject security attributes determined by the ST Author], none*].

b) Destination subject security attributes:

- TOE's network identifier; and
- [*selection: [assignment: other subject security attributes determined by the ST Author], none*].

93 *Application Note: For the subjects, the administrator knows the set of identifiers that can be associated with the physical VPN interfaces; therefore, they are not "presumed" identifiers. The term "identifiers" was used instead of "addresses" to allow for technologies that are not address-based (e.g., circuit identifiers instead of source and destination addresses).*

94 *The ST author should specify other attributes that are used to identify the source and destination subject sets, based on the technology implemented by the TOE.*

c) Information security attributes:

- presumed identity of source subject;
- identity of destination subject;
- transport layer protocol;
- source subject service identifier;
- destination subject service identifier (e.g., TCP or UDP destination port number); and

95 *Application Note: Not all of the above security attributes will exist in all network packets. The intent is that if a network packet includes any of the above security attributes, those attributes will be used in the policy decision. The data link frame type identifies the type of data the data link header encapsulates (e.g., in the case of ARP, the frame type value is 0x0806). The transport layer protocol is what is specified in the 8-bit protocol field in the IP header (e.g., this would include ICMP (value of 1) and is not limited to TCP (value of 6) or UDP (value of 17)). The concept of a "service identifier" may differ depending on the networking stack used; the intent is to specify a service that may exist above the network and transport layers in the protocol stack. A "service" in the IP stack would be NTP, TFTP, etc.*

- [selection: for an IP-based network stack: ICMP message type and code as specified in RFC 792, [selection: [assignment: other information security attributes associated with services identified in FIA_UAU.1(1)], none]; or for a non-IP-based network stack: [assignment: information security attributes]].

96 *Application Note: For an IP-based network stack, the ICMP is to be controlled at the message type and code level. The ST author should fill in the first assignment with the attributes associated with services provided by the TOE that the Security Administrator is able to specify when configuring this policy. If no additional services are specified in FIA_UAU.1(1), the ST author should fill the selection with none. If the TOE uses a non-IP-based network stack, than the ST author makes the second selection and assigns attributes to the services identified in FIA_UAU.1(1).*

FDP_IFF.1.2(2) – **Refinement**: The TSF shall permit an information flow between a **source** subject and **the TOE** via a controlled operation if the following rules hold:

- [the presumed identity of the source subject is in the set of source subject identifiers;
- the identity of the destination subject is the TOE;
- the information security attributes match the attributes in an information flow control policy according to the following algorithm [assignment: algorithm used by the TOE to match information security attributes to information flow control policy].

97 *Application Note: This bullet is dependent on the ST's implementation and may have an assignment of none, if the implementation of this policy does not use the TOE ruleset (e.g., another mechanism is used to control the information flow to/from the TOE).*

FDP_IFF.1.3(2) - The TSF shall enforce the [following rules:

- The TOE shall allow source subjects to access TOE services [*selection: for an IP-based network stack: ICMP, [selection: [assignment: list of other network services provided by the TOE, consistent with FIA_UAU.1], none]; or for non-IP-based network stacks: [assignment: list of network services provided by the TOE, consistent with FIA_UAU.1]*] without authenticating those source subjects; and

- The TOE shall allow the list of services specified immediately above to be enabled (become available to unauthenticated users) or disabled (become unavailable to unauthenticated users)].

98 *Application Note: The intent of this requirement (first bullet) is to allow users to access services such as ICMP Echo (ping) without authentication. However, since some sites may not want to allow this capability, the second bullet was added so that an administrator (see FMT_MOF.1(6)) can restrict the services available.*

99 *The ST author should fill in the assignment in the first bullet with a list of services that the VPN provides that can be accessed without authentication by the user, and make sure that this list is the same as is provided in FIA_UAU.1.1].*

FDP_IFF.1.4(2) - The TSF shall provide the following [the Security Administrator shall have the capability to view all information flows allowed by this information flow control policy before the policy is applied].

100 *Application Note: The intent here is to provide the Security Administrator the capability to see what information flow controls will be applied to the TOE before those controls are activated. This gives the Security Administrator the opportunity to address any errors or unintended TOE interactions with users. In the case of this policy, information flow is between a network device and the TOE.*

FDP_IFF.1.5(2) - The TSF shall explicitly authorize an information flow based on the following rules: [none].

FDP_IFF.1.6(2) - The TSF shall explicitly deny an information flow based on the following rules:

- [The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE is not included in the set of source identifiers for the source subject;

- The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE specifies a broadcast identity;

101 *Application Note: A broadcast identity is one that specifies more than one host on a network. It is understood that the TOE can only know the sub-netting configuration of networks directly connected to the TOE's interfaces and therefore can only be aware of broadcast addresses on those networks.*

- The TOE shall reject requests for access or services where the presumed source identity of the information received by the TOE specifies a loopback identifier; and

- The TOE shall reject requests in which the information received by the TOE contains the route (set of host network identifiers) by which information shall flow from the source subject to the TOE].

5.1.3.5 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.

102 *Application Note: One aspect of this requirement is to ensure packets do not contain residual information that may be used in the padding of a packet.*

5.1.4 Identification and authentication (FIA)

TOE security functions implemented by a probabilistic or permutational mechanism (e.g., password or hash function) are required (at EAL2 and higher) to include a strength of function claim. Strength of Function shall be demonstrated for the authentication mechanism used by the administrator at SOF-medium, as defined in Part 1 of the CC. Specifically, the local authentication mechanism must demonstrate adequate protection against attackers possessing a moderate attack potential.

5.1.4.1 FIA_AFL.1 Authentication failure handling

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 - Refinement: The TSF shall detect when [***an administrator configurable positive*** *integer*] **of** unsuccessful authentication attempts occur related to [administrators attempting to authenticate remotely and authorized IT entities**].**

103 *Application Note: This requirement does not apply to the local administrators, since it does not make sense to lock a local administrator's account in this fashion. This could be addressed by requiring a separate account for local 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.*

104 *Authorized IT enties is intended to address IT enities that are trusted to modify TSF data (e.g., NTP server) or entities that are to be authenticated when establishing an encrypted channel.*

FIA_AFL.1.2 – Refinement: When the defined number of unsuccessful authentication attempts has been met ~~or surpassed~~, the TSF shall **[**at the option of the Security Administrator prevent the remote administrators, or an authorized IT entity from performing activities that require authentication until an action is taken by the Security Administrator, or until a Security Administrator defined time period has elapsed**].**

5.1.4.2 FIA_ATD.1 User attribute definition

FIA_ATD.1.1 – **Refinement**: The TSF shall maintain the following list of security attributes belonging to an **administrator**:

a) [user identifier(s):

- role;
- [selection: [assignment: Any security attributes related to a user identifier (e.g., certificate associated with the userid)], none]; and

b) [selection: [assignment: other user security attributes], none]].

105 *Application Note: This requirement applies to authorized users: administrators and authorized IT entities. The intent is to allow multiple userids to be associated with a user. This allows a single human user to assume multiple roles, albeit requiring authentication as the userid associated with a given role. The intent is for a userid to only be associated with a single role, thus limiting the amount of damage if an administrative role is compromised.*

106 *Item "b" could include the session establishment criteria identified in FTA_TSE depending on the TOE's implementation of the session establishment function.*

5.1.4.3 FIA_UAU.1 Timing of authentication (for TOE-provided services)

FIA_UAU.1.1 - The TSF shall allow [assignment: *list of TOE-provided services*] on behalf of the user to be performed before the user is authenticated.

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

107 *Application Note: The ST writer should fill in the assignment on the list of services provided by the TOE (e.g., ICMP Echo (ping), ARP communications) that are accessible to users without authentication. Users in the context of this requirement and this PP are intended to include external IT entities. The identified services have management constraints identified in FMT_MOF.1.1(4).*

5.1.4.4 FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 – Refinement: The TSF shall require **the Administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of these authorized users.

108 *Application Note: Although FIA_UID.1.2(*) requires all other actions by users to be mediated, this requirement is levied to make the set of users required to authenticate to the TOE clear. Note that the authentication is required only when the specified user is performing a function related to the authentication; for instance, if an administrator wants to utilize an unauthenticated service from the list in FIA_UID.1.1(1), they are not required to authenticate to that service.*

5.1.4.5 FIA_UAU_(EXT).5 Multiple authentication mechanisms

FIA_UAU_(EXT).5.1 - The TSF shall provide a local authentication mechanism, [selection: [assignment: other authentication mechanism(s)], none] to perform user authentication.

109 *Application Note: This extended requirement is needed because there is no CC requirement (other than FIA_UAU.5) that requires the TSF provide authentication (it is implied by other FIA_UAU requirements, but not explicitly required).*

110 *The ST author could chose to fill in the assignment with any additional authentication mechanism such as a single-use authentication mechanism, or a mechanism that authenticates users by using a certificate. If an asymmetric algorithm is chosen, the TOE may rely upon a certificate authority server to obtain a user's certificate, and this server would be considered an authorized IT entity and IT environment requirements should be levied on this IT entity*.

5.1.4.6 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.

111 *Application Note: All users (administrators, users using the TOE-provided services in FIA_UAU.1, and users passing traffic through the VPN) will always be identified at least by a source network identifier. In the case of administrators there will probably be a "userid" as well.*

5.1.4.7 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 **authorized** user.

112 *Application Note: User security attributes are defined in FIA_ATD.1.*

5.1.5 Security management (FMT)

5.1.5.1 FMT_MOF.1(1) Management of security functions behavior (TSF non-Cryptographic Self-test)

FMT_MOF.1.1(1) - The TSF shall restrict the ability to *determine and modify the behavior of* the functions:

- [TSF Self-Test (FPT_TST_(EXT).1)]

to [the Security Administrator].

113 *Application Note: "Invoke" refers to running the self-tests. "Modify the behavior" refers to specifying the interval at which the tests periodically run, or perhaps selecting a subset of the tests to run.*

5.1.5.2 FMT_MOF.1(2) Management of security functions behavior (Cryptographic Self-test)

FMT_MOF.1.1(2) - The TSF shall restrict the ability to *enable, disable* the functions

- [Crypto Self-Test (FPT_TST.1(1), and) Key Generation Self-Test (FPT_TST.1(2)]

to [the Cryptographic Administrator].

114 *Application Note: The enabling or disabling of the cryptographic self-tests immediately after key generation.*

5.1.5.3 FMT_MOF.1(3) Management of security functions behavior (audit and alarms)

FMT_MOF.1.1(3) - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions

- [Security Audit (FAU_SAR)] to [an Administrator].

5.1.5.4 FMT_MOF.1(4) Management of security functions behavior (audit and alarms)

FMT_MOF.1.1(4) - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions

- [Security Audit Analysis (FAU_SAA); and
- Security Audit (FAU_SEL)]

to [the Security Administrator].

115 *Application Note: For the Audit function, enable and disable refer to the ability to enable or disable the audit mechanism as a whole. "Determine the behavior" means the ability to determine specifically what on the system is being audited, while "modify the behavior" means the ability to set or unset specific aspects of the audit mechanism, such as what user behavior is audited, etc.*

5.1.5.5 FMT_MOF.1(5) Management of security functions behavior (audit and alarms)

FMT_MOF.1.1(5) - The TSF shall restrict the ability to *enable, or disable* the functions

- [Security Alarms (FAU_ARP)]

to [the Security Administrator].

116 *Application Note: This requirement ensures only the Security Administrator can enable or disable (turn on or turn off) the alarm notification function – messages and/or the audible alarm. 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 adding, "modify the behavior" to this requirement.*

5.1.5.6 FMT_MOF.1(6) Management of security functions behavior (available TOE-services for unauthenticated users)

FMT_MOF.1.1(6) - The TSF shall restrict the ability to *enable, disable* the functions

- [[selection: for an IP-based network stack: ICMP, [selection: [assignment: other defined services for which authentication is not required in (FIA_UAU.1(1))], none], or for a non-IP-based network stack:[assignment: defined services for which authentication is not required in (FIA_UAU.1(1))]].

to [the Security Administrator].

117 *Application Note: "Enable" refers to allowing a specific service to be specified as being one that is available to users on the network without those users first authenticating to the TOE. "Disable" refers to not allowing such a service to be available. This requirement, coupled with FIA_UAU.1, allows the Security Administrator to specify which TOE services are available to network users without authentication; if they choose, they can completely disable all such services so that unauthenticated users may only attempt to send traffic through the firewall. For the protocol required in FIA_UAU.1, this requirement defines the minimum level of control that must be provided to the Security Administrator. If the ST author provides additional services in FIA_UAU.1, then they should consider specifying the level of control the Security Administrator has with respect to those protocols in this requirement.*

5.1.5.7 FMT_MOF.1(7) Management of security functions behavior (quota mechanism)

FMT_MOF.1.1(7) - The TSF shall restrict the ability to *determine the behavior of* the functions

- [An administrator-specified network identifier;
- set of administrator-specified network identifiers;
- administrator-specified period of time]

to [the Security Administrator].

118 *Application Note: "determine the behavior of" refers to specifying the network identifier(s) that will be tracked using the FRU_RSA.1(2) requirement and the time period over which the quota limitations are enforced. Note that the specification of the actual quotas, while part of the resource allocation functionality, is done by FMT_MTD.2(2).*

5.1.5.8 FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 - The TSF shall enforce the [VPN SFP] to restrict the ability to [manipulate] the security attributes [referenced in the indicated polices] to [an Administrator].

119 *Application Note: The term "manipulate" is used to indicate that the security attributes in FDP_IFF.1.1* may be used to create additional "attributes" that can be used in specifying*

information flow policy rules (for example, a set of IP addresses that can be used as a "group"); this requirement restricts such capabilities to an administrator.

5.1.5.9 FMT_MSA.3(1) Static attribute initialization (attributes)

FMT_MSA.3(1).1 - **Refinement:** The TSF shall enforce the [VPN SFP] to provide *restrictive* default values for the (**security attributes) information flow policy ruleset** that **is** (**are**) used to enforce the SFP.

120 *Application Note: "restrictive" in this case means that by default information is not allowed to flow (according to the referenced policies) unless an explicit rule in the information flow policy ruleset allows an information flow. By default, information is not allowed to flow.*

FMT_MSA.3(1).2 The TSF shall allow the **[**Security Administrator**]** to specify alternative initial values to override the default values when an object or information is created.

121 *Application Note: Since a VPN ruleset typically does not provide multiple initial default values for the rules (that is, there is generally only one "default" rule) this requirement may not apply for all TOEs. In TOEs that allow default values to be specified for individual rules, this requirement indicates that the specification must be done by the Security Administrator.*

5.1.5.10 FMT_MSA.3(2) Static attribute initialization (services)

FMT_MSA.3(2).1 – **Refinement:** The TSF shall enforce the [UNAUTHENTICATED TOE SERVICES SFP] to provide *restrictive* default values for (**security attributes) the set of TOE services available to unauthenticated users (that are used to enforce the SFP).**

122 *Application Note: Since FDP_IFF.1.3(2) allows the TOE to provide services to unauthenticated users, "restrictive" in this case indicates that such services are not available by default, and must be explicitly enabled by the Security Administrator.*

FMT_MSA.3(2).2 The TSF shall allow the [Security Administrator] to specify alternative initial values to override the default values when an object or information is created.

123 *Application Note: This component was used to ensure that no services are provided to unauthenticated users by default, and that the Security Administrator has control over this list of services. FMT_MSA.3.2 (2) might be used to allow the Security Administrator to allow*

a service to be enabled when the TOE is restarted, rather than having the service unavailable by default when the TOE boots up.

FMT_MTD.1(1) Management of TSF data (non-cryptographic, non-time TSF data)

FMT_MTD.1(1) - **Refinement:** The TSF shall restrict the ability to [selection: change default, query, modify, delete, clear, [assignment: other operations] **all** the [TSF data except cryptographic security data and the time and date used to form the time stamps in FPT_STM.1] to [the administrators or authorized IT entities].

124 *Application Note: The ST should iterate this requirement as necessary to ensure that the TSF data are characterized in terms of the functionality provided by the TOE, and that the access is appropriately restricted to administrators. The cryptographic security data and time stamp data are covered in the following two components, as they have specific requirements to support the PP's threats and policies.*

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

FMT_MTD.1.1(2) - The TSF shall restrict the ability to *modify* the [cryptographic security data] to [the Cryptographic Administrator].

125 *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.*

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

FMT_MTD.1.1 - 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 or authorized IT entity].

126 *Application Note: The ST author is able to restrict the ability to set the time and date to the just the Security Administrator, to just an authorized IT entity (e.g., NTP server) or both.*

5.1.5.13 FMT_MTD.1(4) Management of TSF data (VPN Policy Ruleset)

FMT_MTD.1.1(4) – The TSF shall restrict the ability to [*query, modify, delete, create,* [assignment: other operations as specified by the ST Author]] the [VPN Policy rules] to [the Security Administrator].

127 *Application Note: This restricts the specification of the VPN policy ruleset (the SPD and SAD) identified in the FDP_IFF requirements to the administrator. This specification is done using the attributes defined for those policies.*

128 *The ST writer should fill in any TOE-specific operations that an administrator can perform on the ruleset in the assignment.*

5.1.5.14 FMT_MTD.2(1) Management of limits on TSF data (transport-layer quotas)

FMT_MTD.2.1(1) - The TSF shall restrict the specification of the limits for [quotas on transport-layer connections] 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].

129 *Application Note: Note that the wording of FRU_RSA.1(1) does not indicate that the TOE must provide the Security* Administrator *the means to adjust the maximum quota; however, if the TOE does provide such a mechanism then FMT_MTD.2.1(1) would require that that mechanism is restricted to the Security Administrator.*

130 *For FMT_MTD.2.2(1), 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.*

5.1.5.15 FMT_MTD.2(2) Management of limits on TSF data (controlled connection-oriented quotas)

FMT_MTD.2.1(2) - The TSF shall restrict the specification of the limits for [quotas on controlled connection-oriented resources] 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].

131 *Application Note: For FMT_MTD.2.2(2), the ST author should specify the actions that the TOE takes for each controlled connection-oriented resource when the quota (with respect the specific network identifier or set of network identifiers) established by the Security Administrator is reached. This requirement may have to be iterated to be consistent with FRU_RSA.1(2). See the application note on FRU_RSA.1(2) for more detail on the requirements for the quota mechanism.*

5.1.5.16 FMT_REV.1 Revocation

FMT_REV.1.1 –The TSF shall restrict the ability to revoke security attributes associated with the [users, information flow policy ruleset, services available to unauthenticated users**,** [assignment: other resources] within the TSC to [the Security Administrator].

132 *Application Note: The security attributes associated with users are defined in FIA_ATD.1; the intent is to include an indication that a user is allowed to act in a role (Security Administrator, Cryptographic Administrator or Audit Administrator) and an indication that a user is allowed to use an authenticated proxy service.*

133 *The security attributes associated with the information flow policy ruleset are the rules themselves, and any attributes listed in the FDP_IFF.1.1(*) elements that are grouped to create new attributes that can be used in forming a rule.*

134 *The security attributes associated with the services available to unauthenticated users is just the list of services.*

135 *The ST author should specify all other resources that may have "revocable" aspects as implemented in the TOE, and ensure that FMT_REV.1.2 specifies rules for these resources. This list may be empty in an ST.*

136 FMT_REV.1.2 - **Refinement:** The TSF shall **immediately** enforce the:

- [revocation of a user's role (Security Administrator, Cryptographic Administrator, Audit Administrator);
- changes to the information flow policy ruleset when applied;
- disabling of a service available to unauthenticated users;
- changes to the set of security associations with peer TOEs; and
- [selection: [assignment: other rules], none]].

137 *Application Note: The ST author should specify any rules covering additional resources detailed in the assignment in FMT_REV.1.1.*

5.1.5.17 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 - The TSF shall maintain the roles:

- [Security Administrator;
- Cryptographic Administrator (i.e., users authorized to perform cryptographic initialization and management functions);

- Audit Administrator;
- Authorized IT enties; and
- [selection: [assignment: any other roles], none]].

FMT_SMR.2.2 - The TSF shall be able to associate users with roles.

FMT_SMR.2.3 - The TSF shall ensure that the conditions

- [All roles shall be able to administer the TOE locally;
- all roles shall be able to administer the TOE remotely;
- all roles are distinct; that is, there shall be no overlap of operations performed by each role, with the following exceptions:
- all administrators can review the audit trail; and
- all administrators can invoke the self-tests] are satisfied.

Application Note: The administering of the TOE is limited to the capabilities associated with an administrative role. . When the term administrator is used in this PP it refers to a person acting in any of the roles specified in FMT_SMR.2.1. The FIPS 140 validated cryptographic module for this TOE (level 3 for Roles) requires that unique trusted user identifiers be assigned to administer the cryptographic module. Only users associated with the Security Administrator role are allowed to administer the cryptographic module.

5.1.5.18

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions:

- [restrict the ability to invoke determine and modify the behavior of the functions: [TSF Self-Test (FPT_TST_(EXT).1)] to [the Security Administrator;
- restrict the ability to enable, disable the functions TSF Self-Test (Crypto Self-Test (FPT_TST.1(1), and) Key Generation Self-Test (FPT_TST.1(2)) to the Cryptographic Administrator;
- restrict the ability to enable, disable, determine and modify the behavior of the functions Security Audit (FAU_SAR) to an Administrator;
- restrict the ability to enable, disable, determine and modify the behavior of the functions Security Audit Analysis (FAU_SAA); and Security Audit (FAU_SEL) to the Security Administrator;

- restrict the ability to enable, or disable the functions Security Alarms (FAU_ARP) to the Security Administrator

- restrict the ability to enable, disable the functions [[selection: for an IP-based network stack: ICMP, [selection: [assignment: other defined services for which authentication is not required in (FIA_UAU.1(1))], none], or for a non-IP-based network stack:[assignment: defined services for which authentication is not required in (FIA_UAU.1(1))]] to [the Security Administrator].

- restrict the ability to determine the behavior of the functions An administrator-specified network identifier; set of administrator-specified network identifiers; administrator-specified period of time] to [the Security Administrator];

- enforce the [VPN SFP] to restrict the ability to manipulate the security attributes referenced in the indicated polices to an Administrator;

- enforce the [VPN SFP] to provide restrictive default values for the information flow policy rule set security attributes that is used to enforce the SFP;

- enforce the [UNAUTHENTICATED TOE SERVICES SFP] to provide restrictive default values security attributes that are used to enforce the SFP;

- restrict the ability to [selection: change default, query, modify, delete, clear, [assignment: other operations] all the [TSF data except cryptographic security data and the time and date used to form the time stamps in FPT_STM.1] to [the administrators or authorized IT entities];

- restrict the ability to modify the cryptographic security data to the Cryptographic Administrator;

- restrict the ability to set the time and date used to form the time stamps in [FPT_STM.1] to the Security Administrator or authorized IT entity;

- restrict the ability to query, modify, delete, create, [assignment: other operations as specified by the ST Author] the VPN Policy rules to the Security Administrator;

- restrict the specification of the limits for quotas on transport-layer connections to the Security Administrator;

- restrict the specification of the limits for quotas on controlled connection-oriented resources to the Security Administrator;

- [assignment: list of additional security management functions to be provided by the IT environment]].

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 FPT_RCV.1 Manual Recovery

FPT_RCV.1.1 – **Refinement:** After **a** [failure or service discontinuity], the TSF shall enter a maintenance mode where the ability to return **the TOE** to a secure state is provided.

5.1.6.2 FPT_RPL.1 Replay detection

FPT_RPL.1.1 - The TSF shall detect replay for the following entities: [TSF data and security attributes].

FPT_RPL.1.2 - The TSF shall perform: [reject data, audit event and [assignment: list of specific actions]] when replay is detected.

138 *Application Note: Receiving multiple network packets due to network congestion or lost packet acknowledgments is not considered a replay attack. The intent of this requirement is to ensure that an administrative session (in part, in its entirety, by a remote administrator or an authorized IT entity) or a user's authentication sequence cannot be replayed.*

5.1.6.3 FPT_STM.1 Reliable time stamps

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

5.1.6.4 Extended: TSF Testing (FPT_TST_(EXT).1)

FPT_TST_(EXT).1.1 The TSF shall run a suite of self tests during the initial start-up and also either periodically during normal operation, or at the request of an authorized administrator to demonstrate the correct operation of the TSF.

FPT_TST_(EXT).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 .

5.1.6.5 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 F of this profile** during initial start-up **(on power on)**, at the request of the **cryptographic administrator (on demand)**, **under various**

conditions defined in section 4.9.1 of FIPS 140-2, 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 .

5.1.6.6 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]

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

5.1.7 Resource Allocation (FRU)

5.1.7.1 FRU_RSA.1(1) Maximum quotas

FRU_RSA.1.1(1) - **Refinement:** The TSF shall enforce maximum quotas of the following resources: [transport-layer representation] that ***users*** can use *over a specified period of time*.

140 *Application Note: "transport-layer representation" refers specifically to the TCP SYN attack, where half-open connections are established thus exhausting the connection table resource.*

5.1.7.2 FRU_RSA.1(2) Maximum quotas (controlled connection-oriented quotas)

FRU_RSA.1.1(2) – **Refinement:** The TSF shall enforce **administrator-specified** maximum quotas of the following resources: [controlled connection-oriented resources] that ***users associated with an administrator-specified network identifier and a set of administrator-specified network identifiers*** can use *over* ***an administrator****-specified period of time*.

141 *Application Note: This requirement applies to a network entity attempting to exhaust the specified connection-oriented resources (or set of such resources) on the TOE. Connectionless sessions are not a concern because they do not consume resources that persist like connection-oriented sessions do.*

142 *The ST author should fill in the first assignment with the list of connection-oriented resources to which this requirement applies. That is, when a network entity uses such a connection-oriented resource (or a collection of these resources), the TOE tracks that use for the purpose of determining whether the entity has exceed the quota established by the administrator.*

143 *The ST author should use the first selection to indicate whether the TOE is able to track the assignment of the specified resources based on a single network identifier (e.g., a specific IP address) or multiple network identifiers (e.g., a specific IP subnet address). The second selection should reflect the way in which the TOE tracks such resource use. Note that the ST author may have to iterate this requirement if different resources can be controlled differently by the TOE. The ST author should ensure that FMT_MTD.2(2) specifies the actions that are taken for each resource on which there is a quota.*

5.1.8 TOE Access (FTA)

5.1.8.1 FTA_SSL.1 TSF-initiated session locking

FTA_SSL.1.1 – **Refinement:** The TSF shall lock **a local** interactive session after [assignment: **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 - The TSF shall require the following events to occur prior to unlocking the session: [the administrator to re-authenticate].

5.1.8.2 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 - The TSF shall require the following events to occur prior to unlocking the session [the administrator to re-authenticate].

144 *Application Note: The interactive sessions in FTA_SSL.1 and FTA_SSL.2 are those of the local administrator.*

5.1.8.3 FTA_SSL.3 TSF-initiated termination

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

145 *Application Note: A remote session applies to remote administrators, authenticated proxy users, and any connection to a service on the VPN as defined in FDP_IFF.1.3(1)(a).*

5.1.8.4 FTA_TAB.1 Default TOE access banners

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

146 *Application Note: The access banner applies whenever the TOE will provide a prompt for identification and authentication (i.e., administrators). 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 the user of the product and version number).*

5.1.8.5 FTA_TSE.1 TOE session establishment

FTA_TSE.1.1 - **Refinement:** The TSF shall be able to deny establishment **of an administrator session** based on [location, time, and day].

5.1.9 Trusted Path/Channels (FTP)

5.1.9.1 FTP_ITC.1(1) Inter-TSF trusted channel (Prevention of Disclosure)

FTP_ITC.1.1(1) - **Refinement:** The TSF shall **use encryption to** provide a **trusted** communication channel between itself and **authorized IT entities** 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(1) **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.

147 *Application Note: The encryption used to protect the communication channel from disclosure is the symmetric algorithm specified in FCS_COP.1(1).*

148 *FTP_ITC.1.2 is used to ensure secure communications between the TOE and authorized IT entities (e.g., certificate authority server). While these authorized IT entities may initiate communications, it may be the case that the TOE is required to perform a "pull" operation (e.g., obtaining a certificate from a certificate authority, obtaining time from an NTP server).*

FTP_ITC.1.3(1) - The TSF shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: list of other functions for which a trusted channel is required], none]].

149 *Application Note: The "other functions" are the services that are provided by the authorized IT entities (e.g., NTP). FTP_ITC.1(2) Inter-TSF trusted channel (Detection of Modification)*

5.1.9.2 FTP_ITC.1(2) Inter-TSF trusted channel (Detection of Modification)

FTP_ITC.1.1(2) - **Refinement:** The TSF shall **use a cryptographic signature** to provide a **trusted** communication channel between itself and **authorized IT entities** 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.1.2(2) - **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.

150 *Application Note: The method used to provide detection of data modification transmitted through the communication channel is the cryptographic digital signature algorithm specified in FCS_COP.1(2).*

151 *FTP_ITC.1.2 is used to ensure secure communications between the TOE and authorized IT entities (e.g., certificate authority server). While these authorized IT entities may initiate communications, it may be the case that the TOE is required to perform a "pull" operation (e.g., obtaining a certificate from a certificate authority, obtaining time from an NTP server).*

FTP_ITC.1.3(2) - The TSF shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: list of other functions for which a trusted channel is required], none]].

152 *Application Note: The "other functions" are the services that are provided by the authorized IT entities (e.g., NTP).*

5.1.9.3 FTP_TRP.1(1) Trusted path (Prevention of Disclosure)

FTP_TRP.1.1(1) - **Refinement:** The TSF shall provide **an encrypted** communication path between itself and ***remote administrators*** 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.1.2(1) - **Refinement:** The TSF shall permit ***administrators*** to initiate communication via the trusted path.

FTP_TRP.1.3(1) –The TSF shall require the use of the trusted path for all remote administration actions

153 *Application Note: The encryption used to protect the communication channel from disclosure is the symmetric algorithm specified in FCS_COP.1(1)*

154 *"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.*

5.1.9.4 FTP_TRP.1(2) Trusted path (Detection of Modification)

FTP_TRP.1.1(2) - **Refinement:** The TSF shall **use a cryptographic signature to** provide a communication path between itself and ***administrators*** 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.1.2(2) - **Refinement:** The TSF shall permit ***administrators*** to initiate communication via the trusted path.

FTP_TRP.1.3(2) –The TSF shall require the use of the trusted path for all remote administration actions.

155 *Application Note: The method used to provide detection of data modification transmitted through the communication channel is the cryptographic digital signature algorithm specified in FCS_COP.1(2).*

156 *"all administration actions" means that the entire 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.*

5.1.10 Strength of Function Requirement

157 The minimum strength of function level for the security functional requirements is SOF-medium.

5.2 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

158 This Protection Profile provides functional requirements for the IT Environment. The IT environment includes authorized IT entities (e.g., a certificate authority server, NTP 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.

5.2.1 FTP_ITC.1(1) Inter-TSF trusted channel (Prevention of Disclosure)

FTP_ITC.1.1(1) - **Refinement:** The **IT Environment** shall 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(1) - **Refinement:** The **IT Environment** shall permit *the TSF* ***or the IT Environment*** to initiate communication via the trusted channel.

FTP_ITC.1.3(1) - The **IT Environment** shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: communications with authorized IT entities determined by the ST author], none]].

159 *Application Note: If a certificate authority server plays a role in the authentication of users, then the CA is considered an authorized IT entity and the TSF is expected to initiate secure communications with this entity. If the TSF makes use of an NTP server, it is expected that the TSF would initiate the trusted channel with the NTP server.*

5.2.2 FTP_ITC.1(2) Inter-TSF trusted channel (Detection of Modification)

FTP_ITC.1.1(2) - **Refinement:** The **IT Environment** shall provide **an encrypted** 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.1.2(2) - **Refinement:** The **IT Environment** shall permit *the TSF*, or ***the IT Environment*** to initiate communication via the trusted channel.

FTP_ITC.1.3(2) - The **IT Environment** shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: communications with authorized IT entities determined by the ST author], none]].

160 *Application Note: If a certificate authority server plays a role in the authentication of users, then the CA is considered an authorized IT entity and the TSF is expected to initiate secure communications with this entity. If the TSF makes use of an NTP server, it is expected that the TSF would initiate the trusted channel with the NTP server.*

5.2.3 FTP_TRP.1(1) Trusted path (Prevention of Disclosure)

FTP_TRP.1.1(1) - **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 modification or disclosure.

FTP_TRP.1.2(1) - The **IT Environment** shall permit *remote administrators* ***of the TSF*** to initiate communication to the TSF via the trusted path.

FTP_TRP.1.3(1) – **Refinement:** The **IT Environment** shall **initiate** the use of the trusted path for ***all remote administration actions***, [assignment: other services for which trusted path is required].

161 *Application Note: The encryption used to protect the communication channel from disclosure is the symmetric algorithm specified in FCS_COP.1(1).*

162 *This requirement (as is FTP_ITC.1) 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.*

5.2.4 FTP_TRP.1(2) Trusted path (Detection of Modification)

FTP_TRP.1.1(2) - **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 detection of the modification of data.**

FTP_TRP.1.2(2) - **Refinement:** The **IT Environment** shall permit *remote **administrators of the TSF*** to initiate communication **to the TSF** via the trusted path.

FTP_TRP.1.3(2) – **Refinement:** The **IT Environment** shall **initiate** the use of the trusted path for *user authentication, **all remote administration actions**, [selection: [assignment: other services for which trusted path is required]*.

163 *Application Note: The method used to provide detection of data modification transmitted through the communication channel cryptographic signature algorithm specified in FCS_COP.1(2).*

164 *This requirement (as is FTP_ITC.1) 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.*

5.3 TOE SECURITY ASSURANCE REQUIREMENTS

165 This section defines the assurance requirements for the TOE. Table 8 summarizes the components for medium robustness. The augmented requirements are in bold print.

166 The TOE assurance requirements for this PP do not map to a CC EAL. The assurance requirements are summarized in the Table 4 below, with the extended requirements in bold print..

Table 8 Assurance Requirements

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
DEVELOPMENT	ADV_ARC.1	Security Architectural Description

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS 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
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 desceibe 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 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

167 This section provides the rationale for the selection of the IT security requirements, objectives, assumptions, and threats. In particular, it shows that the IT security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment.

6.1 RATIONALE FOR TOE SECURITY OBJECTIVES

168 This section provides a rationale for the existence of each assumption, threat, and policy statement that compose the IDS System Protection Profile. Table 9 demonstrates the mapping between the assumptions, threats, and polices to the security objectives is complete. The following discussion provides detailed evidence of coverage for each assumption, threat, and policy.

Table 9 Rationale for TOE Security Objectives

Threat/Policy	Objectives Addressing the Threat	Rationale
T.ADDRESS_MASQUERADE A user on one interface may masquerade as a user on another interface to circumvent the TOE policy.	O.MEDIATE The TOE must mediate the flow of information between sets of TOE network interfaces or between a network interface and the TOE itself in accordance with its security policy.	O.MEDIATE (FDP_IFC.1(1), FDP_IFF.1(1), (FDP_IFC.1(2), FDP_IFF.1(2)) counters this threat by ensuring that all network packets that flow through the TOE are subject to the information flow policies. One of the rules in FDP_IFF.1(1) ensures that the network identifier in a network packet is in the set of network identifiers associated with a TOE's network interface. Therefore, if a user supplied a network identifier in a packet that was associated with a TOE network interface other than the one the user supplied the packet on, the packet would not be allowed to flow through the TOE, or access TOE services. This would, for example, prevent a user from sending a packet from the Internet claiming to be on a machine on the protected enclave.

Threat/Policy	Objectives Addressing the Threat	Rationale
T.ADMIN_ ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	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.
	O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	O.ADMIN_ROLE (FMT_SMR.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.
	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.MANAGE (FMT_MTD.1(1), FMT_MTD.1(4)) contributes to mitigating this threat by providing administrators the capability to view configuration settings. For example, if the Security Administrator made a mistake when configuring the ruleset, providing them the capability to view the rules affords them the ability to review the rules and discover any mistakes that might have been made.
T.ADMIN_ROGUE An administrator's intentions may become malicious resulting in user or TSF data being compromised.	O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	O.ADMIN_ROLE (FMT_SMR.2) mitigates this threat by restricting the functions available to an administrator. This 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 Security Administrator becomes malicious they could affect the directory access control policy, but the Audit Administrator may be able to detect those actions.

Threat/Policy	Objectives Addressing the Threat	Rationale
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.	O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information.	O.AUDIT_PROTECTION (FAU.SAR.2, FAU_STG.1-NIAP-0429, FAU_STG.3, FAU_STG.NIAP-0414-1-NIAP-0429, FMT_MOF.1(2)) contributes to mitigating this threat by controlling access to the audit trail. No one is allowed to modify audit records, the Audit Administrator is the only one allowed to delete the audit trail. The TOE has the capability to prevent auditable actions from occurring if the audit trail is full.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	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.
	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.SELF_PROTECTION (ADV_ARC.1) contributes to countering this threat by ensuring that the TSF can protect itself from users. 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 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	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION (FCS_CKM.4) 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 module and then reallocated to another process.

Threat/Policy	Objectives Addressing the Threat	Rationale
mechanisms and the data protected by those mechanisms.	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.SELF_PROTECTION (ADV_ARC) contributes to countering this threat by ensuring that the TSF can protect itself from users. 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.DOCUMENT_KEY_LEAKAGE The bandwidth of channels that can be used to compromise key material shall be documented.	O.DOCUMENT_KEY_LEAKAGE (AVA_CCA_(EXT).2) addresses this threat by requiring the developer to perform a 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 A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain access to data or TOE resources.	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.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_ATD.1, FIA_UID.2, FIA_UAU.1, FIA_UAU.2, FIA_UAU_(EXT).5, FTA_TSE.1,AVA_VAN.4) 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 mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.

Threat/Policy	Objectives Addressing the Threat	Rationale
	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.TRUSTED_PATH (FTP_ITC.1(1), FTP_ITC.1(2)) ensures that the communication path end points between the TOE and authorized users (remote administrators, authorized IT entities) are defined. This mechanism allows the TOE to be assured that it is communicating with an authorized user. This also ensures that the transmitted data cannot be compromised or disclosed (e.g., encrypted). The protection offered by this objective is limited to TSF data and security attributes.
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.	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.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 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.SOUND_DESIGN The TOE will be designed using sound design principles and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.	O.SOUND_DESIGN (ADV_FSP_.4, 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,. The design of the TOE can be better understood, which increases the chances that design errors will be discovered.
	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.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.

Threat/Policy	Objectives Addressing the Threat	Rationale
T.FLAWED_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.CHANGE_MANAGEMENT (ALC_CMC.4, ALC_CMS.4, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1,) This objective plays a role in mitigating this threat in the same way that the poor 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.
	O.SOUND_IMPLEMENTATION The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.	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.
	O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.	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.

Threat/Policy	Objectives Addressing the Threat	Rationale
	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.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 TOE behavior being undiscovered.	O.CORRECT_ TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	While these testing activities are a necessary activity 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, Crypto Self-Test (FPT_TST.1(1), and) Key Generation Self-Test (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) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.

Threat/Policy	Objectives Addressing the Threat	Rationale
	O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.	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 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 ensure the TSF satisfies the security functional requirements and demonstrates 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 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) 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.
T.REPLAY A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes (captured as it was transmitted during the course of legitimate use).	O.REPLAY_DETECTION The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.	O.REPLAY_DETECTION (FPT_RPL.1) prevents a user from replaying TSF data and security attributes (e.g., TSF data or security attributes transmitted between a remote administrator, the authentication server, an authorized IT entity and the TOE) that could leave the TOE in a configuration that the administrative staff did not intend (e.g., an administrator modifies the auditable events to be recorded and a user captures that traffic. At a later date the administrator determines that the new set of auditable events is not sufficient and again modifies the events to be audited. The user then replays the earlier audit event configuration.)

Threat/Policy	Objectives Addressing the Threat	Rationale
	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.ROBUST_TOE_ACCESS (FIA_UAU_(EXT).5) contributes to countering this threat by requiring the TOE have the capability to invoke a single-use authentication mechanism. A single-use authentication mechanism ensures that once authentication data has been presented to authenticate a user, that authentication data cannot be used again, therefore a user could not capture authentication and reuse it at a later time.
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.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION (FDP_RIP.2, FCS_CKM.4) 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 will not have residual data from another packet due to the padding of a packet.
T.RESOURCE_EXHAUSTION A malicious process or user may block others from system resources (e.g., connection state tables) via a resource exhaustion denial of service attack.	O.RESOURCE_SHARING The TOE shall provide mechanisms that mitigate attempts to exhaust connection-oriented resources provided by the TOE (e.g., entries in a connection state table; TCP connections used by proxies).	O.RESOURCE_SHARING (FRU_RSA.1(1), FRU_RSA.1(2), FMT_MTD.2(1), FMT_MTD.2(2), FMT_MOF.1(7)) mitigates this threat by requiring the TOE to provide controls over connection-oriented resources. These controls provide the administrator ability to specify which network identifiers have access to the TOE's connection-oriented resources over a time period that is specified by the administrator. 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).

Threat/Policy	Objectives Addressing the Threat	Rationale
T.SPOOFING An entity may misrepresent itself as the TOE to obtain authentication data.	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.	It is possible for an entity other than the TOE (a subject on the TOE, or another IT entity) 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.1(1), FTP_ITC.1(2), FTP_TRP.1(1), FTP_TRP.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 A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.DISPLAY_BANNER The TOE shall display an advisory warning regarding use of the TOE.	O.DISPLAY_BANNER (FTA_TAB.1) helps mitigate this threat by providing the Security 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 attacker with information that may increase their ability to compromise the TOE.
	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.MANAGE (FMT_MTD.1(1)-(4), FMT_MSA.1, FMT_MOF.1(1)-(3)) is necessary because an access control policy is not specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION (FDP_RIP.2, FCS_CKM.4) is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.

Threat/Policy	Objectives Addressing the Threat	Rationale
	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.SELF_PROTECTION (ADV_ARC.1, FTP_TRP.1, FTP_ITC.1) requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed. Without this objective, there could be no assurance that users could not view or modify TSF data or TSF executables. 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.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.TRUSTED_PATH (FTP_TRP.1(1), FTP_TRP.1(2), FTP_ITC.1(1), FTP_ITC.1(2)) plays a role in addressing this threat by ensuring that a trusted communication path exists between the TOE and authorized users (i.e., remote administrators, authorized IT entities). This ensures the transmitted data cannot be compromised or disclosed (e.g., encrypted) during the duration of the trusted path. The protection offered by this objective is limited to TSF data and security attributes (i.e., the data communication between peer TOEs via a VPN is protected by the VPN policy stated in FDP_IFC.1(1) and FDP_IFF.1(1) and FTP_ITC does not apply to VPN communications).

Threat/Policy	Objectives Addressing the Threat	Rationale
T.UNATTENDED_SESSION A user may gain unauthorized access to an unattended session.	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.ROBUST_TOE_ACCESS (FTA_SSL.1, FTA_SSL.2, FTA_SSL.3) 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 Security 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.

Threat/Policy	Objectives Addressing the Threat	Rationale
T.UNAUTHORIZED_ACCESS An unauthorized user may gain access to user or TOE data for which they are not authorized by the security policy.	O.MEDIATE The TOE must mediate the flow of information between sets of TOE network interfaces or between a network interface and the TOE itself in accordance with its security policy.	O.MEDIATE (FDP_IFF.1(1), FDP_IFF.1(2), FDP_IFC.1(1), FDP_IFC.1(2), FMT_REV.1, ADV_ARC.1) works to mitigate this threat by ensuring that all network packets that flow through the TOE are subject to the information flow policies. One of the rules ensures that the network identifier in a packet is in the set of network identifiers associated with a TOE's network interface. Therefore, if a user supplied a network identifier in a packet that purported to originate from a network associated with a TOE network interface other than the one the user supplied the packet on, the packet would not be allowed to flow through the TOE, or access TOE services. The VPN policy ensures that user data being sent between PEER TOEs is encrypted if there is a rule (specified by the Security Administrator) that states data is to be encrypted between those two hosts. The VPN policy allows the administrator to specify for each originating host (identified by IP address), which destination addresses must be accessed through a VPN (using ESP tunnel mode) and which destination addresses may be access without VPN encryption. 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 administrative sessions. If an administrator is not currently accessing the TOE, the message is stored and immediately displayed the next time an administrator accesses the TOE.

Threat/Policy	Objectives Addressing the Threat	Rationale
		The TOE requires successful authentication By implementing strong authentication to gain access to these services, an attacker's opportunity to successfully conduct a man-in-the-middle and/or password guessing attack is greatly reduced. Lastly, the TSF must ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Security Administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy. 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.UNAUTHORIZED_PEER An unauthorized IT entity may attempt to establish a security association with the TOE.	O.PEER_AUTHENTICATION The TOE will authenticate each peer TOE that attempts to establish a security association with the TOE.	O.PEER_AUTHENTICATION (FCS_IKE_(EXT).1) mitigates this threat by requiring that the TOE implement the Internet Key Exchange protocol, as specified in RFC2409, to establish a secure, authenticated channel between the TOE and another remote VPN endpoint before establishing a security association with that remote endpoint.

Threat/Policy	Objectives Addressing the Threat	Rationale
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.	O.AUDIT_REVIEW The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.	O.AUDIT_REVIEW (FAU_SAA.1-NIAP-0407, FAU_ARP.1, FAU_SAR.1, FAU_SAR.3) helps to mitigate this threat by providing the Security Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures, set number of information policy flow failures, self-test failures, etc.) and immediately notifies all TOE administrators 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 administrative sessions. If an administrator is not currently logged into the TOE, the message is stored and immediately displayed the next time an administrator accesses the TOE. This message is displayed to all administrative roles and will remain on the screen for each administrative role until each administrative role acknowledges the message. 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. The TOE can also be configured to generate an audible alarm, which may alert administrators who are not sitting at their administrative workstation or console. The TOE also requires an Audit Administrative 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.

Threat/Policy	Objectives Addressing the Threat	Rationale
T.UNKNOWN_STATE When the TOE is initially started or restarted after a failure, design flaws, improper TOE configurations may cause the security state of the TOE may be unknown.	O.MAINT_MODE The TOE shall provide a mode from which recovery or initial startup procedures can be performed.	O.ROBUST_ADMIN_GUIDANCE (AGD_OPE.1, AGD_PRE.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).
	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.CORRECT_TSF_OPERATION (FPT_TST_(EXT).1, Crypto Self-Test (FPT_TST.1(1), and) Key Generation Self-Test (FPT_TST.1(2)), counters this threat by ensuring that the TSF runs a suite of tests to successfully demonstrates the correct operation of the TSF's underlying abstract machine (hardware and software), the TSF, 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, the Security Administrator can verify the integrity of the TSF's data and stored code as well as the TSF's cryptographic data and stored code.
	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O.MAINT_MODE (FPT_RCV.1) 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 traffic flow and requires an administrator to follow documented procedures to return the TOE to a secure state.

Threat/Policy	Objectives Addressing the Threat	Rationale
	O.CORRECT_ TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	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 security states of the TOE at startup or restart after failure should be documented and understood, thereby reducing the possibility that the TOE's security state could be unknown to users of the TOE.
P.ACCESS_BANNER 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 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 Security Administrator configurable banner that provides all users with a warning about the unauthorized use of the TOE.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	O.AUDIT_GENERATION (FAU_GEN.1-NIAP-0407, FAU_GEN.2-NIAP-0410, FIA_USB.1, FAU_SEL.1-NIAP-0407) addresses this policy by providing the Security Administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).
	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.ROBUST_TOE_ACCESS (FIA_UID.2, FIA_UAU_(EXT).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. While the user ID of authorized users can be assured, since they are authenticated, this PP allows unauthenticated users to access the TOE and the identity is then a presumed network identifier (e.g., IP address).

Threat/Policy	Objectives Addressing the Threat	Rationale
	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.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 Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.
P.ADMIN_ACCESS Administrators shall be able to administer the TOE both locally and remotely through protected communications channels.	O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	O.ADMIN_ROLE (FMT_SMR.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 (e.g., setting the ruleset).
	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.TRUSTED_PATH (FTP_TRP.1(1), FTP_TRP.1(2), FTP_ITC.1(1), FTP_ITC.1(2)) satisfies this policy by requiring that each remote administrative session (all administrative roles) is authenticated and conducted via a secure channel. Additionally, all authorized IT entities (e.g. authentication/certificate servers, NTP servers) must adhere to the same requirements as the remote administrator.
P.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.	O.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.	O.CRYPTOGRAPHIC_FUNCTIONS 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.

Threat/Policy	Objectives Addressing the Threat	Rationale
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).	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.CRYPTOGRAPHY_VALIDATED satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused.	O.RESIDUAL_INFORMATION satisfies this policy by ensuring that cryptographic data are cleared from resources that are shared between users. Keys must be zeroized according to FIPS 140-2.
P.INTEGRITY The TOE shall support the IETF *Internet Protocol Security Encapsulating Security Payload* (IPSEC ESP) as specified in RFC 2406. Sensitive information transmitted to a peer TOE shall apply integrity mechanisms as specified in *Use of HMAC-SHA-1-96 within ESP and AH* (RFC 2404).	O.INTEGRITY The TOE must be able to protect the integrity of data transmitted to a peer TOE via encryption and provide IPSec authentication for such data. Upon receipt of data from a peer TOE, the TOE must be able to decrypt the data and verify that the received data accurately represents the data that was originally transmitted.	O.INTEGRITY (FDP_IFC.1(1), FDP_IFF.1(1)) satisfies this policy by ensuring that all IPSEC encrypted data received from a peer TOE is properly decrypted and authentication verified.
P.VULNERABILITY_ ANALYSIS_TEST The TOE must undergo analysis and testing by the NSA to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.	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.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

169 The purpose for the environmental objectives is to provide protection for the TOE that cannot be addressed through IT measures. The defined objectives provide for physical protection of the TOE and proper management of the TOE. Together with the IT security objectives, these environmental objectives provide a complete description of the responsibilities of TOE in meeting security needs.

170 All but one of the security objectives for the environment, OE.CRYPTANALYTIC, are restatements of an assumption found in Section 3. Therefore, those security objectives for the non-IT environment trace to the assumptions trivially and are suitable for covering the assumptions.

171 The IT environment security objective OE.CRYPTANALYTIC is necessary to play a role in countering the threat T.CRYPTO_COMPROMISE. This IT environment security objective ensures that the cryptographic methods used in the IT environment are interoperable with the mechanisms provided by the TOE. The IT environment's cryptographic methods should be independently validated to be FIPS 140-2 compliant. OE.CRYPTANALYTIC maps to the IT environmental iterated requirements FPT_ITC.1 (ensuring that encryption is used on the communication channel between authorized IT entities and the TOE), and FPT_TRP (ensuring that an administrator and authenticated proxy users can be assured that they are communicating with the TOE).

6.3 RATIONALE FOR TOE SECURITY REQUIREMENTS

172 This section demonstrates that the functional components selected for the IDS System Protection Profile provide complete coverage of the defined security objectives. The mapping of components to security objectives is depicted in the Table 10.

Table 10 Requirements vs. Objectives Mapping

Objective	Requirements Addressing the Objective	Rationale

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_ROLE The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	FMT_SMR.2	FMT_SMR.2 requires that three roles exist for administrative actions: the Security Administrator, who is responsible for configuring the TOE's security policies; the Cryptographic Administrator, who is restricted to managing the security data that is critical to the cryptographic operations; and the Audit Administrator, who is restricted to reading the audit trail. 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. The functionality of the roles, as defined by this PP, is predicated on the notion that once the TOE has been setup and is running in a stable configuration the Security Administrator would not be required to frequently administer the TOE. The Audit Administrator will probably be logging into the TOE most often to review the audit trail. Restricting the Audit Administrator's capabilities thus reduces the potential harm that could occur due to an error, or the execution of malicious code.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	FAU_GEN.1-NIAP-0407 FAU_GEN.2-NIAP-0410 FIA_USB.1 FAU_SEL.1-NIAP- 0407 FAU_STG.3 FAU_STG-NIAP-0414-1-NIAP-0429	FAU_GEN.1-NIAP-0407 defines the set of events that the TOE must be capable of recording. This requirement ensures that the Security 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. FAU_GEN.2-NIAP-0410 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the userid. In all other cases, the association is based on the source network identifier, which is presumed to be the correct identity, but cannot be confirmed since these subjects are not authenticated. FAU_SEL.1-NIAP-0407 allows the Security Administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.

Objective	Requirements Addressing the Objective	Rationale
		FAU_STG.3 requires that the administrators be alerted when the audit trail exceeds a capacity threshold established by the Security Administrator. This ensures that the Security Administrator has the opportunity to manage the audit trail before it becomes full and the avoiding the possible loss of audit data. 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 Security Administrator or Audit Administrator) that would generate an audit record (e.g., depending on the FAU_SEL.1-NIAP-0407 configuration, traffic may no longer flow through the TOE) or the audit mechanism will overwrite the oldest audit records with new records. 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.
O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information.	FMT_MOF.1(2) FAU_SAR.2 FAU_STG.1-NIAP-0429 FAU_STG.3 FAU_STG.NIAP-0414-1-NIAP-0429	FAU_SAR.2 restricts the ability to read the audit trail to the Audit Administrator, 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). 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 Security Administrator or if the option of overwriting old audit records is chosen by the Security Administrator in FAU_STG.NIAP-0414-1-NIAP-0429, the TOE audit data may be deleted. This helps ensure that audit records are kept until the Security Administrator deems they are no longer necessary. This requirement also ensures that no one has the ability to modify audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the audit trail is maintained. FMT_MOF.1(2) restricts the capability to modify the behavior of the audit and alarm functions to the Security Administrator. While the Audit Administrator has the capability to choose how they will review the audit trail, they do not have the capability to select what events are audited. This requirement ensures that only the Security Administrator can turn audit on or off, thus ensuring user's actions are audited according to a site defined

Objective	Requirements Addressing the Objective	Rationale
		policy.
O.AUDIT_REVIEW The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.	FAU_ARP.1 FAU_ARP_ACK_(EXT).1 FAU_SAA.1-NIAP-0407 FAU_SAR.1 FAU_SAR.3 FMT_MOF.1(3) FMT_MOF.1(4) FMT_MOF.1(5)	FAU_ARP.1 requires that the alarm be displayed at the local administrative console and at the remote administrative console(s) when an administrative session exists. For the latter, the alarm is sent to each role either during an established session or upon session establishment. This is required to ensure that no matter which role an administrator accesses, the alarm will be received as soon as possible. This 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 because the Audit Administrator is the only administrative role that can explicitly read the audit trail. If the Security Administrator were the first administrative role to receive the alarm it would be unacceptable for them not to have access to specific details (e.g., VPN rule that was violated, source network identifier of the entity that caused the alarm) concerning the event that fired the alarm. FAU_ARP_ACK_(EXT).1 requires that the alarm be displayed at the local administrative console until it is acknowledged by an administrator, and at the remote administrative console(s) until it has been acknowledged by an administrator acting in each of the administrative roles. This ensures that the alarm message will not be obstructed and the administrators will be alerted of a potential security violation. FAU_SAA.1-NIAP-0407 defines the events that indicate a potential security violation and will generate an alarm. The triggers for these events are configurable, for the most part, by the Security Administrator. The exception is that any failure of the TSF self-tests will generate an alarm. FAU_SAR.1 provides the Audit Administrator with the capability to read all the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the Audit Administrator to interpret the audit trail, which is subject to interpretation. It is expected that the audit information be presented in such a way that the Audit Administrator can examine an audit record and have the appropriate information

Objective	Requirements Addressing the Objective	Rationale
		(that required by FAU_GEN.2-NIAP-0410) presented together to facilitate the analysis of the audit review.
		FAU_SAR.3 complements FAU_SAR.1 by providing the Audit Administrator the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the Audit Administrator be able to establish the audit review criteria based on a userid and source subject identity, so that the actions of a user can be readily identified and analyzed. The criteria also includes a destination subject identity so the Audit Administrator can determine what network traffic is destined for an individual machine. Allowing the Audit Administrator to perform searches or sort the audit records based on dates, times, subject identities, destination service identifier, or transport layer protocol provides the capability to extract the network activity to what is pertinent at that time in order facilitate the Audit Administrator's review. Being able to search on the destination service identifier affords the Audit Administrator the opportunity to see what traffic is destined for a service (e.g., TCP port) or set of services regardless of where the traffic originated. It is important to note that the intent of sorting in this requirement is to allow the Audit Administrator the capability to organize or group the records associated with a given criteria. For example, if the Audit Administrator wanted to see what network traffic was destined for the set of TCP ports 1-1024, they would be able to have the audit data presented in such a way that all the traffic for TCP port 1 was grouped together, all the traffic for port 2 was grouped together and so on. FMT_MOF.1(3) restricts the ability to control the behavior of the audit and alarm mechanism to the administrators. The Security Administrator is the only user that controls the behavior of the events that generate alarms. FMT_MOF.1(4) provides the administrators "read only" access to the audit records and prohibits access to all other users. Additionally the administrators are provided the capability to "search and sort" audit on defined criteria. This capability expedites problem resolution analysis. FMT_MOF.1(5) ensures that only an administrators can "enable or disable" the security alarms. This

Objective	Requirements Addressing the Objective	Rationale
		requirement works with FMT_MOF.1(4) to provide detailed granularity to the administrator when determining which actions constitute a security violation
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.	ALC_CMC.4 ALC_CMS.4 ALC_DVS.1 ALC_FLR.2 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. 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 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.

Objective	Requirements Addressing the Objective	Rationale
		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.
		This aids in understanding how the CM system enforces the control over changes made to the TOE.
O.CORRECT_ TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	FPT_TST_(EXT).1, FPT_TST.1(1) FPT_TST.1(2)	O_CORRECT_TSF_OPERATION requires three security functional requirements in the FPT class, FPT_TST. These functional requirements provide the end user with the capability to ensure the TOE's security mechanisms continue to operate correctly in the field. FPT_TST_(EXT).1 has been created to ensure end user tests exist to demonstrate the correct operation of the security mechanisms required by the TOE that are provided by the hardware and that the TOE's software and TSF data has not been corrupted. 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. Crypto Self-Test (FPT_TST.1(1), and) Key Generation Self-Test (FPT_TST.1(2) are necessary to ensure the correctness of the TSF software and TSF data for crypto functions. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies.
O.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.	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	The FCS requirements used in this PP satisfy this objective by levying requirements that ensure the cryptographic standards include the NIST FIPS publications (where possible) and NIST approved ANSI standards. The intent is to have the satisfaction of the cryptographic standards be validated through a NIST FIPS 140 validation. In contrast to O.CRYPTOGRAPHY_VALIDATED, this objective is to provide cryptographic functionality that is used by the TOE. The core functionality to be supported is encryption/decryption using a symmetric algorithm, and digital signature generation and verification using asymmetric algorithms. Since these operations involve cryptographic keys, how the keys are generated and/or otherwise obtained have to also be

Objective	Requirements Addressing the Objective	Rationale
		specified. FCS_CKM.1(1) is a requirement that a cryptomodule generate symmetric keys. Such keys are used by the AES encryption/decryption functionality specified in FCS_COP.1(1). FCS_CKM.1(2) is a requirement that a cryptomodule generate asymmetric keys. Such keys are used for cryptographic signatures as specified in FCS_COP.1(2). FCS_CKM.1 requires that the TSF validate all keys generated to assure that meet relavant standards. FCS_CKM_(EXT).2 requires that keys are handled appropriately and associated with the correct entities, and that transfer of keys is done with error detection. Storage of persistent secret and private keys must be done in a secure fashion. FCS_COP.1(3) requires that the TSF provide hashing services using a NIST-approved implementation of the Secure Hash Algorithm. Another way of obtaining key material for symmetric algorithms is through cryptographic key establishment, as specified in FCS_COP.1(4). Key establishment has two aspects: key agreement and key distribution. Key agreement occurs when two entities exchange public data yet arrive at a mutually shared key without ever passing that key between the two entities (for example, the Diffie-Hellman algorithm). Key distribution (FCS_CKM.2) occurs when the key is transmitted from one entity to the TOE. If the entity is electronic and a protocol is used to distribute the key, it is referred to in this PP as “Key Transport”. If the key is loaded into the TOE it can be loaded electronically (e.g., from a floppy drive, smart card, or electronic keyfill device) or manually (e.g., typed in). One or more of these methods must be selected. FCS_CKM.4 provides the functionality for ensuring key and key material is zeroized. This applies not only to key that resides in the TOE, but also to intermediate areas (physical memory, page files, memory dumps, etc.) where key may appear. As previously mentioned FCS_COP.1(1) specifies that AES be used to perform encryption and decryption operations. FCS_COP.1(2) gives three options for providing the digital signature capability; these requirements reference the approvirate standards for each digital signature option.

Objective	Requirements Addressing the Objective	Rationale
		FCS_COP_(EXT).1 requires that any random number generation and hashing functions, respectively, are part of a FIPS-validated cryptographic module. This requirement does not mandate that the functionality is generally available, but only that it be implemented in a FIPS-validated module should other cryptographic functions need these services.
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.	FCS_BCM_(EXT).1 FCS_CKM.1(1) FCS_CKM.1(2)	This objective deals with the issue of using FIPS 140-2-approved cryptomodules in the TOE. A cryptomodule, as used in the components, is a module that is FIPS 140-2 validated (in accordance with FCS_BCM_(EXT).1); the cryptographic functionality implemented in that module are FIPS-approved security functions that have been validated; and the cryptographic functionality is available in a FIPS-approved mode of the cryptomodule. This objective is distinguished from O.CRYPTOGRAPHIC_FUNCTIONS in that this deals only with a requirement to use FIPS 140-2-validated cryptomodules where the TOE requires such functionality; it does not dictate the specific functionality that is to be used. FCS_BCM_(EXT).1 is an extended requirement that specifies not only that cryptographic functions that are FIPS-approved must be validated by FIPS, but also what NIST FIPS rating level the cryptographic module must satisfy. The level specifies the degree of testing of the module. The higher the level, the more extensive the module is tested. FCS_CKM.1(1) and FCS_CKM.1(2) mandates that the cryptomodule must generate key, and that this key generation must be part of the FIPS-validated cryptomodule.
O.DISPLAY_BANNER 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 Security Administrator defined banner before a user can establish an authenticated session. This banner is under complete control of the Security 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 The bandwidth of channels that can be used to compromise key material shall be documented.	AVA_CCA_(EXT).t	AVA_CCA_(EXT).1 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

Objective	Requirements Addressing the Objective	Rationale
		keys should be in order to reduce the damage due to a key compromise.
O.INTEGRITY The TOE must be able to protect the integrity of data transmitted to a peer TOE via encryption and provide IPSec authentication for such data. Upon receipt of data from a peer TOE, the TOE must be able to decrypt the data and verify that the received data accurately represents the data that was originally transmitted.	FDP_IFC.1(1) FDP_IFF.1(1)	O.INTEGRITY (FDP_IFC.1(1), FDP_IFF.1(1)) satifies this policy by ensuring that all IPSEC encrypted data received from a peer TOE is properly decrypted and authentication verified.
O.MAINT_MODE The TOE shall provide a mode from which recovery or initial startup procedures can be performed.	FPT_RCV.1	This objective is met by using the FPT_RCV.1 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 enter a mode where it can no longer be assured of enforcing its security policies. Therefore, the TOE enters a state that disallows traffic flow 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 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 FMT_MSA.3 (1) FMT_MSA.3 (2) FMT_MOF.1(1) FMT_MOF.1(2) FMT_MOF.1(3) FMT_MOF.1(4) FMT_MOF.1(5) FMT_MOF.1(6)	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. FMT_MSA.1 provides the Security Administrator the capability to manipulate the security attributes to facilitate the construction of the ruleset. An example of this would be to group a set of service identifiers that are to have

Objective	Requirements Addressing the Objective	Rationale
	FMT_MOF.1(7) FMT_MTD.1(1) FMT_MTD.1(2) FMT_MTD.1(3) FMT_MTD.1(4) FAU_SAR.1 FAU_SAR.2 FAU_SAR.3 FAU_SEL.1-NIAP-0407 FAU_STG.1-NIAP-0429 FAU_STG.3 FAU_STG.NIAP-0414-1-NIAP-0429 FAU_ARP_ACK_(EXT).1	the same rule applied, rather than having to specify a separate rule for each service identifier. FMT_MSA.3 (1) requires that by default, the TOE does not allow an information flow, rather than allowing information flows until a rule in the ruleset disallows it. FMT_MOF.1(2) and FMT_MSA.3 (2) are related to the services provided by FAU_UAU.1(1) and provide the Security Administrator control as to the availability of these services. FMT_MOF.1(2) provides the ability to enable or disable the TOE services to the Security Administrator. FMT_MSA.3 (2) requires that the these services by default are disabled. Since the Security Administrator must explicitly enable these services it ensures the Security Administrator is aware that they are running. This requirement does afford the Security Administrator the capability to override this restrictive default and allow the services to be started whenever the TOE reboots or is restarted. FMT_MOF.1(1) is used to ensure the administrators have the ability to invoke the TOE self-tests at any time. The ability to invoke the self-tests is provided to all administrators. The Security Administrator is able to modify the behavior of the tests (e.g., select when they run, select a subset of the tests). FMT_MOF.1(3) specifies the ability of the administrators to control the security functions associated with audit and alarm generation. The ability to control these functions has been assigned to the appropriate administrative roles. FMT_MOF.1(7) This requirement limits the ability to manipulate the values that are used in the FRU_RSA.1(2) requirements to the Security Administrator. The Security Administrator is provided the capability to assign the network identifier(s) they wish to place resource restrictions on and allows them to also specify over what period of time those quota limitations are in place. FMT_MOF.1(4) provides the administrators "read only" access to the audit records and prohibits access to all other users. Additionally the administrators are provided the capability to "search and sort" audit on defined criteria. This capability expedites problem resolution analysis.

Objective	Requirements Addressing the Objective	Rationale
		FMT_MOF.1(5) ensures that only an administrators can "enable or disable" the security alarms. This requirement works with FMT_MOF.1(5) to provide detailed granularity to the administrator when determining which actions constitute a security violation FMT_MOF.1(6) limits the ability to enable or disable unauthenticated TOE services for both IP based networks and non-IP based networks to the Security Administrator. These TOE services would be available to appropriate network users at the discretion of the Security Administrator. FMT_MOF.1(7) provides the Security Administration configuration control of the allocation of connection-oriented TOE resources. This requirement provides the Security Administrator with a capability to thwart possible external "resource allocation" attacks on the TOE. 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 requirements. This is necessary because the ST author may add TSF data in assignments that cannot be addressed apriori by the PP authors. 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 an authorized IT entity. It is important to allow this functionality, due to clock drift and other circumstances, but the capability must be restricted. An authorized 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) provides the Security Administrator the capability to manage the TOE's ruleset. This capability is restricted to only the Security Administrator and allows them to create, view, modify and delete the rules that

Objective	Requirements Addressing the Objective	Rationale
		comprise the ruleset. FAU_SAR.1 ensures that the Audit Administrator has the capability to review the audit records and that they are presented in a manner that is suitable for review (e.g., the Audit Administrator can construct a sequence of events provided the necessary events were audited). FAU_SAR.2 restricts the ability to read the audit records to the administrators. This capability exists for the Security and Crypto administrators to help facilitate any trouble shooting that they may have to perform. FAU_SAR.3 provides the administrators with the ability to selectively review the contents of the audit trail based on established criteria. This capability allows the administrators to focus their audit review to what is pertinent at that time. FAU_STG.1-NIAP-0429 specifies that only the Audit Administrator can delete the audit trail. This prevents the accidental or intentional deletion of the audit trail by administrators acting in another role. FAU_STG.3 provides the Security Administrator the capability to establish a threshold of audit trail capacity, that when reached an alarm will be generated. If the audit trail becomes full FAU_STG.NIAP-0414-1-NIAP-0429 provides the Security Administrator the option of having the TOE prevent auditable events from occurring, or having the TOE overwrite the oldest audit records. While the option of overwriting old audit records does not technically prevent audit data loss, it is provided to the Security Administrator as an option to prevent a possible denial-of-service. FAU_ARP_ACK_(EXT).1 contributes to this objective in that it requires the administrators to acknowledge an alarm before it is no longer displayed. Without this requirement an alarm display message may be overwritten or lost without an administrator being aware of the alarm condition.

Objective	Requirements Addressing the Objective	Rationale
O.MEDIATE The TOE must mediate the flow of information between sets of TOE network interfaces or between a network interface and the TOE itself in accordance with its security policy.	FDP_IFF.1(1) FDP_IFF.1(2) FDP_IFC.1(1) FDP_IFC.1(2) FMT_REV.1 ADV_ARC.1	The FDP_IFF and FDP_IFC requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place. FDP_IFC.1(1), and FDP_IFC.1(2) define the subjects, information (e.g., objects) and the operations that are performed with respect to the three information flow policies. FDP_IFC.1(1), the subjects are the TOE's network interfaces. The objects are defined as the network IP packets on which the TOE performs VPN operations. As packets enter the TOE, the network interface where they are received is the source subject. As packets are sent out of the TOE the network interface that they are sent out of is the destination subject. Subjects must be defined as entities that the TOE has control over. The TOE has control over its own network interfaces such that it can make information flow decisions to allow/disallow network packets to flow from in incoming interface to an outgoing interface, and can apply VPN operations to packets that are allowed to flow. To define subjects as the senders and receivers of network packets would not allow specification of an information flow policy that the TOE could enforce, since the sender and receiver of network packets are not under the contol of the TOE. The operations defined are those of the VPN policy. The VPN policy either passes information unmodified, sends encrypted and authenticated packets to a peer TOE, or decrypts and verifies authentication of packets received from a peer TOE. FDP_IFF.1(1) specifies the attributes on which VPN information flow decisions are made. Each TOE interface has a set of source subject identifiers that is the list of senders of information packets that are allowed to send packets to this TOE intercface. Each TOE interface also has a list of desitnation subject identifiers that specifies the reveivers that network packets can be sent to on that TOE interface. As packets are received on a particular network interface, the TOE determines if they are allowed to enter on that interfce. Then based on rules defined by the Security Administrator, the TOE applies VPN operations to the packet. Before the packet is sent out of a particular network interface, the TOE determines if the destination (i.e., receiver) of the packet is in the

Objective	Requirements Addressing the Objective	Rationale
		list of destinations that may be reached over that interface. FDP_IFC.1(2) defines subjects for the unauthenticated access to any services the TOE provides. This is different from the other policies in that the TOE mediates access to itself, rather than determining if information should be allowed to flow through the TOE. The destination subject is defined to be the TOE, and the source subject is the TOE interface on which a network packet is received. The information remains the same, a network packet, and the operations are limited to accept or reject the packet. FDP_IFF.1(2) provides the rules that apply to the unauthenticated use of any services provided by the TOE. ICMP is the only service that is required to be provided by the TOE, and the security attributes associated with this protocol allow the Security Administrator to specify what degree the ICMP traffic is mediated (i.e., the ICMP message type and code). The ST author could specify other services they wish their TOE implementation to provide, and if they do so, they should also specify the security attributes associated with the additional services. FMT_REV.1 is a management requirement that affords the Security Administrator the ability to immediately revoke user's ability to send network traffic to or through the TOE. If the Security Administrator revokes a user's access (e.g., via a rule in the ruleset, revoking an administrative role from a user, revoking a user's ability to use a proxy) the TOE will immediately enforce the new Security Administrator defined "policy". 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. This will ensure that packets that flow through the TOE, or those that are destined for the TOE are mediated with respect to the identified policies. Each TSF interface that operates on subjects or objects that are identified in the explicit policies, or operates on TSF data or security attributes, must ensure that

Objective	Requirements Addressing the Objective	Rationale
		the operation is checked against the explicit and implicit security policies defined in this PP. If any TSF interface allows unchecked access to any of these resources, then the TOE cannot be relied upon to enforce the security policies.
O.PEER_AUTHENTICATION The TOE will authenticate each peer TOE that attempts to establish a security association with the TOE.	FCS_IKE_(EXT).1	The O.PEER_AUTHENTICATION objective is satisfied by the requirement FCS_IKE_(EXT).1, which specifies that the TOE must implement the Interenet Key Exchange protocol defined in RFC 2409. By implementing this protocol, the TOE will establish a secure, authenticated channel with each peer TOE for purposes of establishing a security association, which includes the establishment of a cryptographic key, algorithm and mode to be used for all communication. It is possible to establish multiple security associations between two peer TOEs, each with its own cryptographic key. Authentication may be via a digital signature or pre-shared key.
O.REPLAY_DETECTION The TOE will provide a means to detect and reject the replay of TSF data and security attributes.	FPT_RPL.1	The O.REPLAY_DETECTION objective is satisfied by the requirement FPT_RPL.1, which requires the TOE to not only detect, but to also reject the attempted replay of TSF data (other than authentication data, which is performed by the authentication server that has the FPT_RPL.1 requirement levied upon it in the IT environment), and security attributes. This requirement also requires the TOE to audit the detection of replay, which affords the administrators the opportunity to be aware of users attempting to replay critical data and affect the TOE's ability to enforce security policies as desired by the administrators.
O.RESIDUAL_INFORMATIO N The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	FDP_RIP.2 FCS_CKM.4	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 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). FCS_CKM.4 applies to the destruction of cryptographic keys used by the TSF. This requirement specifies how and when cryptographic keys must be destroyed. The

Objective	Requirements Addressing the Objective	Rationale
		proper destruction of these keys is critical in ensuring the content of these keys cannot possibly be disclosed when a resource is reallocated to a user.
O.RESOURCE_SHARING The TOE shall provide mechanisms that mitigate attempts to exhaust connection-oriented resources provided by the TOE (e.g., entries in a connection state table; TCP connections used by proxies).	FRU_RSA.1(1) FRU_RSA.1(2) FMT_MTD.2(1) FMT_MTD.2(2) FMT_MOF.1 (7)	While an availability security policy does not explicitly exist, FRU_RSA.1 was used to mitigate potential resource exhaustion attempts. FRU_RSA.1(1) was used to reduce the impact of an attempt being made to exhaust the transport-layer representation (e.g., attempt to make the TSF unable to respond to connection-oriented requests, such as SYN attacks). This requirement allows the administrator to specify the time period in which when maximum quota (which is defined by the ST) is met or surpassed, an ST defined action is to take place, which is specified in FMT_MTD.2(1). These two requirements together help limit the resources that can be utilized by the general population of users as a whole. An issue with treating all the users the same is that legitimate users may not be able to establish connections due to the connection table entries being exhausted. Therefore FRU_RSA.1(2) is also included. FRU_RSA.1(2) is more specific in that attempts to exhaust the connection-oriented resources by a single network address, or a set of network addresses can be controlled. This affords the administrator a finer granularity of control than FRU_RSA.1(1). FRU_RSA.1(2) has the advantage of providing the Security Administrator with the ability to define the maximum number of resources a particular address or set of addresses can use over a specified time period. This requirement works in conjunction with FMT_MTD.2(2) which restricts the ability to set the quotas to the security administrator and allows for the ST author to assign what actions will take place once the quotas are met or surpassed. This iteration of FPT_RSA.1 makes it less likely that a legitimate user of the TOE will be denied access due to resource exhaustion attempts. FMT_MOF.1(7) restricts the ability to assign the single network address or set of network addresses used in FRU_RSA.1(2) to the Security Administrator. This is in keeping with the

Objective	Requirements Addressing the Objective	Rationale
		TOE's notion of the Security Administrator is responsible for configuring the TOE's policy enforcement mechanisms.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 AGD_PRE.1 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. 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. 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).

Objective	Requirements Addressing the Objective	Rationale
		Since the non-administrative users of this TOE are limited to proxy users it is expected that the user guidance would discuss the secure use of proxies and how the single-use authentication mechanism is used. The use of the single-use authentication mechanism would not have to be repeated in the administrator's guide. AGD_OPE.1 ensures that the guidance documentation is complete and can be followed unambiguously to ensure the TOE is not mis-configured in an unsecure state due to confusing guidance.
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	FTA_TSE.1 FIA_UID.2 FTA_SSL.1 FTA_SSL.2 FTA_SSL.3 AVA_VAN.4 FIA_AFL.1 FIA_ATD.1 FIA_UAU.1 FIA_UAU.2 FIA_UAU_(EXT).5	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., a user attempting to send a data packet through the TOE that does not require authentication; in which case the identity is presumed to be authentic). In other cases (e.g., proxy users, administrators, and authorized IT entities), the identity of the user is authenticated. It is impractical to require authentication of all users that attempt to send data through the TOE, therefore, the requirements specified in the TOE require authentication where it is deemed necessary. This does impose some risk that a data packet was sent from an identity other than specified in the data packet. FIA_ATD.1 defines the attributes of users, including a userid that is used to by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume). This requirement allows a human user to have more than one user identity assigned, so that a single human user could assume all the roles necessary to manage the TOE. In order to ensure a separation of roles, this PP requires a single role to be associated with a user id. This is inconvenient in that the administrator would be required to log in with a different user id each time they wish to assume a different role, but this helps mitigate the risk that could occur if an administrator were to execute malicious code.

Objective	Requirements Addressing the Objective	Rationale
		FIA_UAU.1 contributes to this objective by limiting the services that are provided by the TOE to unauthenticated users. Management requirements and the unauthenticated information flow policy requirement provide additional control on these services. FIA_UAU.2 was refined since only the VPN only requires that administrators, authorized IT entities and proxy users authenticate themselves to the TOE before performing administrative duties (including those performed by authorized IT entities (e.g., NTP server)), or using the proxy services identified in this requirement. Unlike the unauthenticated proxies, these proxies require authentication, which provides a level of control on who can access the proxies and reduces the potential risk to the TOE. In order to control logical access to the TOE an authentication mechanism is required. The extended requirement FIA_UAU_(EXT).5 mandates that the TOE provide a local authentication mechanism. This requirement also affords the ST author the opportunity to add additional authentication mechanisms (e.g., single-use, certificates) if they desire. 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. FTA_TSE.1.1 contributes to this objective by limiting a user's ability to logically access the TOE. This requirement provides the Security Administrator the ability to control when (e.g., time and day(s) of the week) and where (e.g., from a specific network address) remote administrators, as well as proxy users and

Objective	Requirements Addressing the Objective	Rationale
		authorized IT entities can access the TOE. FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by remote administrators, authenticated proxy users and authorized IT entities. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data by locking the targeted account until the Security Administrator takes some action (e.g., re-enables the account) or for some Security Administrator defined time period. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. 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 Security 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 a Security Administrator defined time interval of inactivity remote sessions will be terminated, this includes user proxy sessions and remote administrative sessions. This component is especially necessary, since remote sessions are not typically afforded the same physical protections that local sessions are provided.

Objective	Requirements Addressing the Objective	Rationale
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.	ADV_ARC.1 FTP_ITC.1(1), FTP_ITC.1(2) FTP_TRP.1(1), FTP_TRP.1(2)	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. FTP_ITC.1(1), FTP_ITC.1(2) and FTP_TRP.1(1), FTP_TRP.1(2) are necessary for communication between the TOE and other trusted IT entities (e.g., authentication server, authorized IT entities) and the TOE and remote administrators. In order to protect TSF data and security attributes there is need for a trusted channel/trusted path. The trusted channel ensures that the authentication data that is supplied to the TOE is not compromised. It may be the case that the TOE relies upon an authorized IT entity to supply/manage TSF data (e.g., time stamp). If this is the case, the trusted channel ensures the TSF data is not compromised. The aspect of the trusted path that applies to this objective is FTP_TRP.1.3, which requires that the entire remote administrative session be protected. The protection of the communication path when TSF data is being transmitted is critical to the TSF maintaining a domain of execution that cannot be tampered or interfered with, thus resulting is a possible unauthorized disclosure or security policy failure.
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.	ADV_ARC.1 ADV_FSP.5 ADV_TDS.4 ADV_INT.1	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. 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

Objective	Requirements Addressing the Objective	Rationale
		security architecture description will demonstrate that the TSF protects itself from tampering and that the TSF prevents bypass of the SFR-enforcing functionality. 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. 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

Objective	Requirements Addressing the Objective	Rationale
		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).!, 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, any application level) 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.
		The design, as required by ADV_TDS.4 , provides the reader 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,

Objective	Requirements Addressing the Objective	Rationale
		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.
O.SOUND_IMPLEMENTATION The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.	ADV_IMP.1 ADV_TDS.4 ADV_INT.1 ALC_TAT.1	While ADV_TDS.4 is used to aide in ensuring that the TOE's design is sound, it also contributes to ensuring the implementation is correctly realized from the design. It is expected that evaluators will use the design as an aide in understanding the implementation representation. The design requirements ensure the evaluators have enough information to intelligently analyze (e.g., the documented interface descriptions of the modules match the entry points in the module, error codes returned by the functions in the module are consistent with those identified in the documentation) the implementation and ensure it is consistent with the design. While evaluators have the ability to "negotiate" the subset in ADV_IMP.1, was chosen to ensure 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. 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

Objective	Requirements Addressing the Objective	Rationale
		of the implementation.
		ALC_TAT.1 provides evaluators with information necessary to understand the implementation representation and what the 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. ADV_TDS.4 also is used here to provide the correspondence of the lowest level of decomposition (e.g., source code) to the adjoining level, low-level design. The correspondence analysis is used by the evaluator as a tool when determining if the design is correctly reflected in the implementation representation.
O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.	ATE_COV.2 ATE_FUN.1 ATE_DPT.3 ATE_IND.2	In order to satisfy O.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

Objective	Requirements Addressing the Objective	Rationale
		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.TIME_STAMPS 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 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. 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, authorized IT entity, or both, depending on the selection made by the ST author. The authorized IT entity was included as an option for the possible use of an NTP server to set the TOE’s time.
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.	FTP_ITC.1(1), FTP_ITC.1(2) FTP_TRP.1(1), FTP_TRP.1(2)	FTP_TRP.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 or modification. 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.1.2) mechanism they can be assured they are communicating with the TOE. FTP_TRP.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

Objective	Requirements Addressing the Objective	Rationale
		authentication functions. Furthermore, the remote administrator's communication path is encrypted during the entire session. FTP_ITC.1(1) and FTP_ITC.1(2) are similar to FTP_TRP.1(1) and FTP_TRP.1(2), in that they require a mechanism that creates a distinct communication path with the same characteristics, however FTP_ITC.1(1) and FTP_ITC.1(2) is used to protect communications between IT entities, rather than between a human user and an IT entity. FTP_ITC.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. Two iterations of FTP_ITC and two iterations of FTP_TRP were necessary to ensure that the trusted channel/path will prevent disclosure, via encryption, as well as detect of modifications, via cryptographic signature. Both iteration will be implemented to ensure that communication is with an authorized IT entity and protected from unauthorized disclosure/modification.
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.	AVA_VAN.4	To maintain consistency with the overall assurance goals of this TOE, O.VULNERABILITY_ANALYSIS_TEST requires the AVA_VAN.4 component to provide the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. 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

173 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 Global Information Grid (GIG);
- Department of Defense (DoD) Instruction 8500.1; and
- the postulated threat environment.

174 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 10.

175 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 Department of Defense (DoD) minimum and require good assurance security design as specified in Evaluation Assurance Level (EAL)3 or greater.

176 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.

177 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 Satisfying All Dependencies

178 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 11 identifies the functional requirement, its correspondent dependency and the analysis and rationale for not supporting the dependency in this PP.

Table 11 Requirement Dependencies

Component	Dependencies	Satisfied
FAU_ARP.1	FAU_SAA.1	FAU_SAA.1 is satisfied by FAU_SAA.1-NIAP-0407.
FAU_SAR.1	FAU_GEN.1	
FCS_CKM.1	FCS_CKM.2	The extended requirement FCS_CKM.1(1) AND FCS_CKM.1(2)were chosen instead of FCS_CKM.2 to more clearly state the requirements as they apply to FIPS 140-2. Therefore, FCS_CKM.1(1) AND FCS_CKM.1(2)satisfies the dependency.
FCS_CKM.1 FCS_CKM.4	FMT_MSA.2	This dependency is satisfied by placing strict requirements on the values of attributes of the cryptographic module in the associated FCS requirements. Therefore, FMT_MSA.2 is not necessary to satisfy the requirement of only secure values being assigned to secure attributes.
FMT_MOF.1 FMT_MSA.1 FMT_MTD.1	FMT_SMF.1	The requirements FMT_MOF.1, FMT_MSA.1, FMT_MTD.1 express the functionality required by the TSF to provide the specified functions to manage TSF data, security attributes, and management functions. These requirements make clear that the TSF has to provide the functions to manage the identified data, attributes, and functions.
FIA_UAU.1 FIA_UAU.2 FMT_SMR.2	FIA_UID.1	This dependency is satisfied with the inclusion of requirement FIA_UID.2. This requirement is hierarchical to FIA_UID.1 and is sufficient to satisfy the dependency for these requirements.
FMT_MOF.1 FMT_MSA.1 FMT_MTD.2	FMT_SMR.1	This dependency is satisfied with the inclusion of requirement

Component	Dependencies	Satisfied
FMT_REV.1		FMT_SMR.2. This requirement is hierarchical to FMT_SMR.1 and is sufficient to satisfy the dependency for these requirements.
FTA_SSL.1 FTA_SSL.2	FIA_UAU.1	This dependency is satisfied with the inclusion of requirement FIA_UAU.2. This requirement is hierarchical to FIA_UAU.1 and is sufficient to satisfy the dependency for these requirements.

6.6 RATIONALE FOR EXTENDED REQUIREMENTS

179 Table 12 presents the rationale for the inclusion of the extended functional and assurance requirements found in this PP. The extended requirements that are included as NIAP interpretations do not require a rationale for their inclusion per CCEVS management.

Table 12 Rationale for Extended Requirements

Extended Requirement	Identifier	Rationale
FAU_ARP_ACK_(EXT).1	Security alarm acknowledgement	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).
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.

Extended Requirement	Identifier	Rationale
FCS_CKM_(EXT).2	Cryptographic Key Handling and Storage	This extended requirement is necessary since the CC does not provide a means to specify a cryptographic key handling and storage implementation.
FCS_COP_(EXT).1	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
FCS_IKE_(EXT).1	Internet Key Exchange	This extended requirement is necessary since the CC does not include requirements for this specific key enchange protocol. This protocol is specified in RFC 2409, but there are specific configurable setting that must be specified that are documented in the extended requirement.
FIA_UAU_(EXT).5	Multiple authentication mechanisms	This extended requirement is needed for local administrators because there is no CC requirement that requires the TSF provide authentication. Because this PP allows the IT environment to provide an authentication server to be used for the single-use authentication mechanism for remote users, it is important to specify that the TSF provide the means for local administrator authentication in case the TOE cannot communicate with the authentication server.

Extended Requirement	Identifier	Rationale
FPT_TST_(EXT).1	TSF testing	This extended requirement is necessary to capture the notion of the TOE to verify the integrity of the TSF software. Additionally, the TSF data set that is subject to these tests was reduced to address the notion that it does not make sense to test the integrity of some TSF data (e.g., audit data) and this extended requirement address that.
AVA_CCA_(EXT).1	Systematic Cryptographic Module Covert Channel Analysis	The intent of the PP authors is to require covert channel analysis only on the cryptographic module(s) and the CC does not have requirements to perform partial covert channel analysis on TOE. AVA_CCA_(EXT).1 provides covert channel analysis only upon the cryptographic module in search for leakage of critical cryptographic security parameters.

7 APPENDICES

A. References

[1] *Common Criteria for Information Technology Security Evaluation,* CCIB-98-031 Version 2.1, August 1999.

[2] *Department of Defense Chief Information Officer Guidance and Policy Memorandum No.6-8510*, Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG), June 2000.

[3] *U.S. Department of Defense Virtual Private Network (VPN) Boundary Gateway Protection Profile for Basic Robustness Environments (BRE),* Version .6, September 2001.

[4] *U.S. Government Firewall Protection Profile for Medium Robustness Environments*, Version 1.0, October 28, 2003.

[5] *Federal Information Processing Standard Publication (FIPS-PUB) 197*, Advanced Encryption Standard (AES*), November 2001*.

[6] *Federal Information Processing Standard Publication (FIPS-PUB) 140-2*, Security Requirements for Cryptographic Modules*,* May 25, 2001.

[7] *Federal Information Processing Standard (FIPS-PUB) 46-3*, Data Encryption Standard (DES), October 1999.

[8] Internet Engineering Task Force, *IP Encapsulating Security Payload (ESP),* RFC 2406, November 1998.

[9] Internet Engineering Task Force*, Internet Key Exchange (IKE),* RFC 2409, November 1998.

[10] *Information Assurance Technical Framework,* Release 3.0, September 2000.

[11] Internet Engineering Task Force, *ESP CBC-Mode Cipher Algorithms,* RFC 2451, November 1998.

[12] Internet Engineering Task Force, *Use of HMAC-SHA-1-96 within ESP and AH,* RFC 2404, November 1998.

[13] *Department of Defense Instruction, Information Assurance Implementation,* 8500.2, February 6,2003.

[14] *Department of Defense Directive, Information Assurance,* 8500.1, October 24,2002.

[15] *Implementing Virtual Private Networks,* Steven Brown, McGraw Hill, 1999.

[16] *A Goal VPN Protection Profile for Protecting Sensitive Information,* Release 2.0, July 2000.

[17] *The AES Cipher Algorithm and Its Use with Ipsec <draft-ietf-ipsec-ciph-aes-cbc.03.txt>*, Internet draft, November 2001

[18] *U.S. Government Protection Profile for Single-Level Operating Systems in Environments Requiring Medium Robustness*, Version 1.67, 30 October 2003

B. Glossary

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[3] and the disclosure and modification of data.[4]

Accountability -- Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator -- A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

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.

Authentication -- Security measure that verifies a claimed identity.

Authentication data -- Information used to verify a claimed identity.

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.

---

[3] Hardware and software.

[4] Stored or communicated.

Availability -- Timely[5], reliable access to IT resources.

Compromise -- Violation of a security policy.

Confidentiality -- A security policy pertaining to disclosure of data.

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 who 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 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 [7]:

- 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 data authentication code computed from data.

Cryptographic Module -- 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.

---

[5] According to a defined metric.

Defense-in-Depth (DID) -- A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

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 certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

DMZ -- A Demilitarized Zone (DMZ) is a network that is mediated by the TOE but, as a result of less stringent access controls, provides access to publicly available services, such as web servers.

Embedded Cryptographic Module -- One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

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.

Entity -- A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

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.

Identity -- A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity -- A security policy pertaining to the corruption of data and TSF mechanisms.

Integrity label -- A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.

Integrity level -- The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Mandatory Access Control (MAC) -- A means of restricting access to objects based on subject and object sensitivity labels.[6]

---

[6] The Bell LaPadula model is an example of Mandatory Access Control

Mandatory Integrity Control (MIC) -- A means of restricting access to objects based on subject and object integrity labels.

Multilevel -- The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object[7] -- 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.

Operating Environment -- The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) -- 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.

---

[7]The only named objects in this PP, are operating system controlled files.

Operational key -- Key intended for protection of operational information or for the production or secure electrical transmissions of key streams.

Peer TOEs -- Mutually authenticated TOEs that interact to enforce a common security policy.

Public Object -- An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

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.

Secure State -- Condition in which all TOE security policies are enforced.

Security attributes -- TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

Security level -- The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity on the information [10].

Sensitivity label -- A security attribute that represents the security level of an object and that describes the sensitivity (e.g. Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions [10].

Split key -- A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.

Subject -- An entity within the TSC that causes operations to be performed.

Symmetric key -- A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

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.

User -- Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability -- A weakness that can be exploited to violate the TOE security policy.

C. Acronyms

AES	Advanced Encryption Standard
BRE	Basic Robustness Environment
CAN	Campus Area Network
CC	Common Criteria for Information Technology Security Evaluation
CM	Configuration Management
CSP	Critical Security Parameters
DAC	Discretionary Access Control
DES	Data Encryption Standard
DID	Defense In Depth
DMZ	Demilitarized Zone
DoD	Department of Defense
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
EBST&S	Enclave Boundary Security Technologies and Solutions
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS PUB	Federal Information Processing Standard Publication
FTP	File Transfer Protocol
GIG	Global Information Grid
HTTP	Hypertext Transfer Protocol
I&A	Identification and Authentication
IATF	Information Assurance Technical Framework
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPSEC ESP	Internet Protocol Security Encapsulating Security Payload
IT	Information Technology
LAN	Local Area Network
MAC	Mandatory Access Control
MAN	Metropolitan Area Network
MIC	Mandatory Integrity Control
MRE	Medium Robustness Environment

NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OS	Operating System
PKI	Public Key Infrastructure
PP	Protection Profile
rDSA	RSA Digital Signature Algorithm
RNG	Random Number Generator
SFP	Security Function Policy
SOF	Strength of Function
SPD	Security Policy Database
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Function
TSFI	TOE Security Function Interface
TSP	TOE Security Policy
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
WAN	Wide Area Network

D. Robustness Environment Characterization

General Environmental Characterization

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: **value of the resources** and **authorization of the entities** to those resources.

In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e. the TOE itself and all of the data processed by the TOE).

Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. In the next section 1.2.2, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

VALUE OF RESOURCES

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the using organization. For example, in the DoD low-value data might be equivalent to data marked "FOUO", while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. 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 firewall may have "low value" data itself, but it might protect an enclave with high value data. If the firewall was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

AUTHORIZATION OF ENTITIES

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router, non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an OS, an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database).

It is important to note that authorization **does not** refer to the **access** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees was authorized to certain data on a TOE, yet they connect the TOE to the Internet. There are millions of entities that are not **authorized** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g., confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

SELECTION OF APPROPRIATE ROBUSTNESS LEVELS

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

When assessing any environment with respect to Information Assurance the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. 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 the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

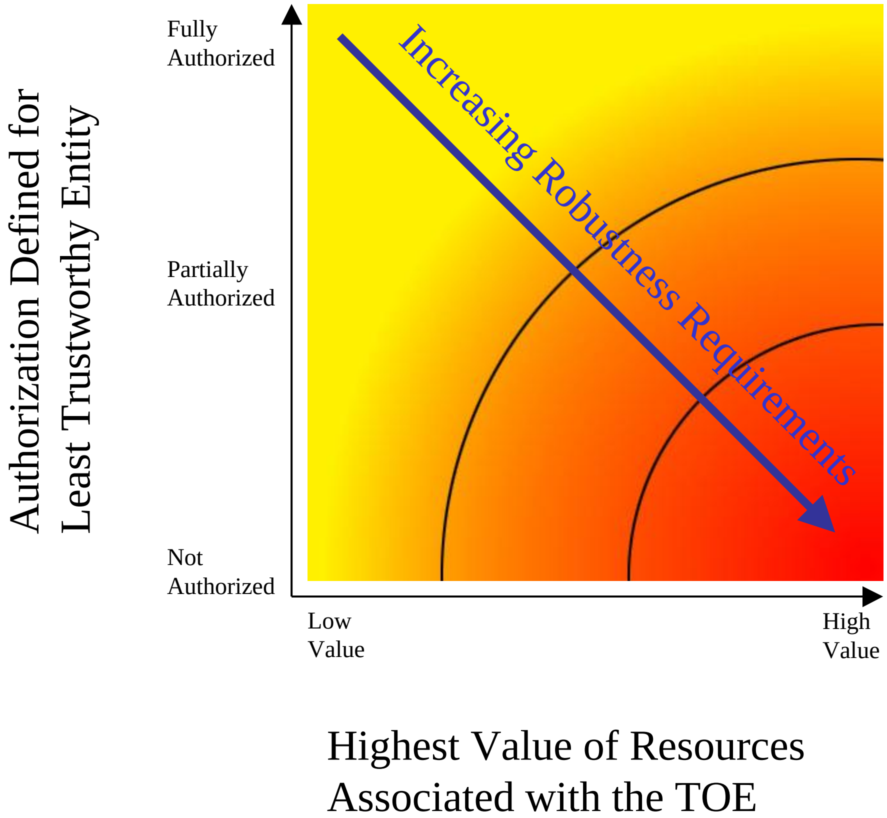
The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g. non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to use the TOE. In this case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again, selection of a basic robustness TOE would be appropriate.

The preceding examples demonstrated that it is possible for radically 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.

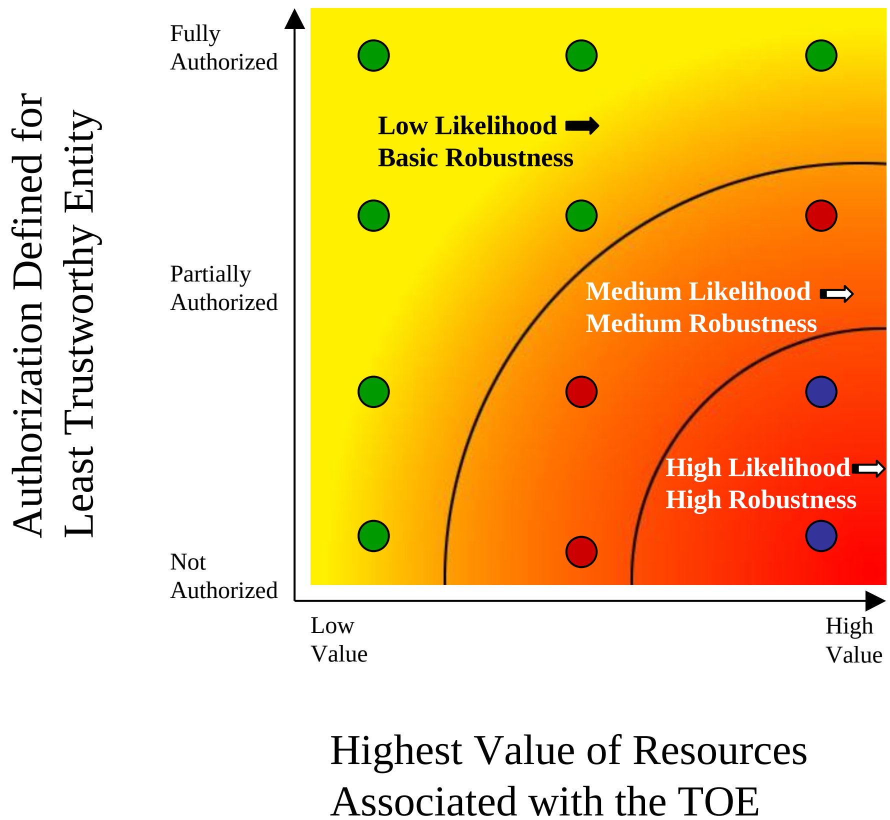
As depicted in the following figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflect 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 a set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in the following chart.


Authorization Defined for
Least Trustworthy Entity
Fully
Authorized
Partially
Authorized
Not
Authorized
Increasing Robustness Requirements
Low
Value
High
Value
Highest Value of Resources
Associated with the TOE

In this second representation of environments and the robustness plane below, 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. In <PP Section>[8] of this PP, the targeted threat level for a medium robustness TOE is characterized. This information is provided to help organizations using this PP ensure that the functional requirements specified by this medium robustness PP are appropriate for their intended application of a compliant TOE.



[8] The PP author should insert the section of the PP that describes the TOE Environment.

E. Refinements

This section contains refinements where text was omitted. Omitted text is shown as bold text within parenthesis. The actual text of the functional requirements as presented in Section 5 has been retained.

FAU_ARP.1.1 – **Refinement:** The TSF shall (**take**) [immediately display an alarm 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:

f) local console,

g) remote administrator sessions that exist, and;

h) remote administrator sessions that are initiated before the alarm has been acknowledged, and;

i) at the option of the Security Administrator, generate an audible alarm, and;

j) [[selection: [assignment: other methods determined by the ST author], "no other methods"]]]

upon detection of a potential security violation.

FAU_GEN.1.1-NIAP-0407 – The TSF shall be able to generate an audit record of the following auditable events:

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

c) All auditable events **(for the)** [listed in Table 7] **(level of audit; and)**

d) [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].

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

FAU_SAA.1.2-NIAP-0407 **- Refinement:** The TSF shall enforce the following rules for monitoring audited events:

((a accumulation or combination of)

[a) Security Administrator specified number of authentication failures;

b) Security Administrator specified number of Information Flow policy violations by an individual presumed source network identifier (e.g., IP address) within an administrator specified time period;

c) Security Administrator specified number of Information Flow policy violations to an individual destination network identifier within an administrator specified time period;

d) Security Administrator specified number of Information Flow policy violations to an individual destination subject service identifier (e.g., TCP port) within an administrator specified time period;

e) Security Administrator specified Information Flow policy rule, or group of rule violations within an administrator specified time period;

f) Any detected replay of TSF data or security attributes;

g) Any failure of the cryptomodule self-tests (FPT_TST.1(1));

h) Any failure of the Key Generation self-tests (FPT_TST.1(2));

i) Any failure of the other TSF self-tests (FPT_TST_(EXT).1);

j) Security Administrator specified number of encryption failures;

k) Security Administrator specified number of decryption failures;

l) Security Administrator specified number of Phase 1 authentication failures when negotiating the Internet Key Exchange protocol;

m) Security Administrator specified number of failures occur during Phase 2 negotiation; and

n) [selection: [assignment: any other rules], "no additional rules"]]

known to indicate a potential security violation;

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

FAU_SAR.2.1 – **Refinement**: The TSF shall prohibit all users read access to the audit records **in the audit trail**, except **(those users that have been granted explicit read-access) the Administrators**.

FAU_STG.3.1 - **Refinement**: The TSF shall **(take)** [immediately alert the administrators by displaying a message at the local console, and at the remote administrative console when an administrative session exists for each of the defined administrative roles, at the option of the Security Administrator generate an audible alarm, [selection: [assignment: other methods], no other methods]] if the audit trail exceeds [a Security Administrator settable percentage of storage capacity].

FAU_STG.3.1 - **Refinement**: The TSF shall (**take)** [immediately alert the administrators by displaying a message at the local console, and at the remote administrative console when an administrative session exists for each of the defined administrative roles, at the option of the Security Administrator generate an audible alarm, [selection: [assignment: other methods], no other methods]] if the audit trail exceeds [a Security Administrator settable percentage of storage capacity].

FCS_CKM.1.1 **Refinement**: The (**TSF) cryptomodule** shall generate **symmetric** cryptographic keys (**in accordance with a specified cryptographic key generation algorithm)** [using a FIPS-Approved Random Number Generator**] (and specified cryptographic key sizes**) [for all key sizes] that meet the following: [one of the standards defined in Annex C to FIPS 140-2].

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

a) [The Key Zeroization Requirements in FIPS PUB 140-2 Key Management Security Levels 3;

b) Zeroization of all private cryptographic keys, plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete; and

c) The zeroization shall be executed by overwriting the key/critical cryptographic security parameter storage area three or more times with an alternating pattern.

d) The TSF shall overwrite each intermediate storage area for private cryptographic keys, plaintext cryptographic keys, and all other critical security parameters three or more times with an alternating pattern upon the transfer of the key/CSPs to another location.]

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with the **NIST-approved digital signature** algorithm ***[selection:***

(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater,

(2) RSA Digital Signature Algorithm (rDSA with odd e) with a key size (modulus) of 2048 bits or greater, or

(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]

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 cryptographic signature, elliptic curves will be required within a TBD time frame after all the necessary standards and other supporting information are fully established.

that meets the following:

a) Case: Digital Signature Algorithm

FIPS PUB 186-2, Digital Signature Standard, for signature creation and verification processing; and ANSI Standard X9.42-2001, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography for generation of the domain parameters;

b) Case: RSA Digital Signature Algorithm (with odd e)

ANSI X 9.31-1998 (May 1998), Digital Signatures Using Reversible Public Key Cryptography For The Financial Services Industry (rDSA);

c) Case: Elliptic Curve Digital Signature Algorithm

ANSI X9.62-1-xxxx (10 Oct 1999), Public Key Cryptography for the Financial Services Industry: Elliptic Curve Digital Signature Algorithm (ECDSA).

FCS_COP.1.1(4) **Refinement:** The TSF shall perform **cryptographic key agreement services** in accordance with a **NIST-approved implementation of a key agreement** algorithm ***[selection:***

(1) Finite Field-based key agreement algorithm and cryptographic key sizes(modulus) of 2048 bits or greater,

(2) Elliptic Curve-based key agreement algorithm and cryptographic key size of 256 bits or greater]

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 within a TBD time frame after all the necessary standards and other supporting information are fully established.

that meets the following:

a) Case: Finite field-based key agreement schemes

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

Application Note: For example, "Classic" Diffie-Hellman-based schemes

b) Case: Elliptic curve-based key agreement schemes

ANSI X9.63-200x (1 Oct 2000), Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography.

Application Note: Some authentication mechanism on the keying material is recommended. In addition, repeated generation of the same shared secrets should be avoided. As an example, the MQV schemes described in the above standards address these issues.

FDP_IFF.1.2(1) **- Refinement:** The TSF shall permit an information flow between a **source** (**controlled)** subject and **a destination subject** (**controlled information)** via a controlled operation if the following rules hold:

- **[**the presumed identity of the source subject is in the set of source subject identifiers;
- the identity of the destination subject is in the set of source destination identifiers;
- the information security attributes match the attributes in an information flow policy rule (contained in the information flow policy ruleset defined by the Security Administrator) according to the following algorithm [assignment: algorithm used by the TOE to match information security attributes to information flow policy rules]; and
- the selected information flow policy rule specifies that the information flow is to be permitted].

FDP_IFF.1.2(3) – **Refinement**: The TSF shall permit an information flow between a **source** (**controlled)** subject and (**controlled information) the TOE** via a controlled operation if the following rules hold:

- [the presumed identity of the source subject is in the set of source subject identifiers;
- the identity of the destination subject is the TOE;

- the information security attributes match the attributes in an information flow control policy according to the following algorithm [assignment: algorithm used by the TOE to match information security attributes to information flow control policy].

FIA_AFL.1.2– **Refinement:** When the defined number of unsuccessful authentication attempts has been met (**or surpassed)**, the TSF shall [at the option of the Security Administrator prevent the remote administrators, or an authorized IT entity from performing activities that require authentication until an action is taken by the Security Administrator, or until a Security Administrator defined time period has elapsed].

FIA_ATD.1.1 – **Refinement**: The TSF shall maintain the following list of security attributes belonging to (**individual users**) **an administrator**:

a) [user identifier(s):

   - role;
   - [selection: [assignment: Any security attributes related to a user identifier (e.g., certificate associated with the userid)], none]; and

b) [selection: [assignment: other user security attributes], none]].

FMT_MSA.3.1 (1) – **Refinement:** The TSF shall enforce the [VPN SFP] to provide *restrictive* default values for the (**security attributes) information flow policy ruleset** that **is** (**are**) used to enforce the SFP.

FMT_MSA.3.1 (2) – **Refinement:** The TSF shall enforce the [UNAUTHENTICATED TOE SERVICES SFP] to provide *restrictive* default values for (**security attributes) the set of TOE services available to unauthenticated users (that are used to enforce the SFP).**

180 FMT_REV.1.2 - **Refinement:** The TSF shall **immediately** enforce the (**rules**):

- [revocation of a user's role (Security Administrator, Cryptographic Administrator, Audit Administrator);
- changes to the information flow policy ruleset when applied;
- disabling of a service available to unauthenticated users;
- changes to the set of security associations with peer TOEs; and
- [selection: [assignment: other rules], none]].

FRU_RSA.1.1(1) – **Refinement:** The TSF shall enforce maximum quotas of the following resources: [transport-layer representation] that **(selection: *individual user, defined group of users, subjects]) users*** can use **([selection: simultaneously])** *over a specified period of time*.

FRU_RSA.1.1(2) – **Refinement:** The TSF shall enforce **administrator-specified** maximum quotas of the following resources: [controlled connection-oriented resources] that that **(selection: *individual user, defined group of users, subjects]) users associated with an administrator-specified network identifier and a set of administrator-specified network identifiers*** can use **([selection: simultaneously])** *over* ***an administrator****-specified period of time*.

FTA_SSL.1.1 – **Refinement:** The TSF shall lock **a local** interactive session after [assignment: **(time of interval of user inactivity) 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 following events to occur) the administrator to re-authenticate** prior to unlocking the session**(: [assignment:** ***events to occur*****].)**

FTA_SSL.2.2 - **Refinement:** The TSF shall require the **(following events to occur) administrator to re-authenticate** prior to unlocking the session**(: assignment: events to occur].)**.

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

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

FTA_TSE.1.1 - **Refinement:** The TSF shall be able to deny **(session)** establishment **of an administrator session** based on [location, time, and day].

FTP_ITC.1.1(1) - **Refinement:** The TSF shall **use encryption to** provide a **trusted** communication channel between itself and (**a remote trusted IT product) authorized IT entities** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from (**modification or)** disclosure.

FTP_ITC.1.2(1) **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities, (the remote trusted IT product)*** to initiate communication via the trusted channel.

FTP_ITC.1.1(2) - **Refinement:** The TSF shall **use a cryptographic signature** to provide a **trusted** communication channel between itself and (**a remote trusted IT product) authorized IT entities** that is logically distinct from other communication channels and provides assured identification of its end points and and (**protection of the channel data from modification or disclosure) detection of the modification of data**.

FTP_ITC.1.2(2) - **Refinement:** The TSF shall permit *the TSF*, ***or the authorized IT entities***, ***(the remote trusted IT product)*** to initiate communication via the trusted channel.

FTP_TRP.1.1(1) - **Refinement:** The TSF shall provide **(a) an encrypted** communication path between itself and ***remote administrators, (remote, local)*** **(users)** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **(modification or)** disclosure.

FTP_TRP.1.2(1) - **Refinement:** The TSF shall permit ***administrators (the TSF, local users, remote users)*** to initiate communication via the trusted path.

FTP_TRP.1.3(1) – **Refinement:** The TSF shall require the use of the trusted path for **all remote administration actions**, [selection: ***(initial user authentication),*** [assignment: other services for which trusted path is required, none]].

FTP_TRP.1.1(2) - **Refinement:** The TSF shall **use a cryptographic signature to** provide a communication path between itself and ***administrators, (remote, local)*** **(users)** that is logically distinct from other communication paths and provides assured identification of its end points and (**protection of the communicated data from) detection of the** modification **(or disclosure) of data**.

FTP_TRP.1.2(2) - **Refinement:** The TSF shall permit ***administrators (the TSF, local users, remote users)*** to initiate communication via the trusted path.

FTP_TRP.1.3(2) – **Refinement:** The TSF shall require the use of the trusted path for **all remote administration actions**, [selection: ***(initial user authentication),*** [assignment: other services for which trusted path is required, none]].

FTP_ITC.1.1(1) - **Refinement:** The (**TSF**) **IT Environment** shall provide a **trusted** communication channel between itself and **(a remote trusted IT product) 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 **(modification or)** disclosure.

FTP_ITC.1.2(1) - **Refinement:** The (**TSF**) **IT Environment** shall permit *the TSF*(***, the remote trusted IT product), or the IT Environment*** to initiate communication via the trusted channel.

FTP_ITC.1.3(1) - **Refinement:** The (**TSF**) **IT Environment** shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: communications with authorized IT entities determined by the ST author], none]].

FTP_ITC.1.1(2) - **Refinement:** The (**TSF**) **IT Environment** shall provide (**a**) **an encrypted** communication channel between itself and **(a remote trusted IT product) 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 modification or disclosure) detection of the modification of data.**

FTP_ITC.1.2(2) - **Refinement:** The (**TSF**) **IT Environment** shall permit *the TSF*(***, the remote trusted IT product)*** **or** ***the IT Environment*** to initiate communication via the trusted channel.

FTP_ITC.1.3(2) - **Refinement:** The (**TSF**) **IT Environment** shall initiate communication via the trusted channel for [all authentication functions, [selection: [assignment: communications with authorized IT entities determined by the ST author], none]].

F. Statistical Random Number Generator Tests

181 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.)

<u>The Monobit Test</u>:

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.

<u>The Poker Test</u>:

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.

<u>The Runs Test</u>:

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.

Table C.1 - Required Intervals for Length of Runs Test

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 Test:

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)** during initial start-up **(on power on)**, at the request of the **cryptographic administrator (on demand)**, **under various conditions,** and periodically **(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**.