U.S. Government Protection Profile

Intrusion Detection System –

System

For

Medium Robustness Environments

Information
Assurance
Directorate

Version 1.1
18 June 2007

Protection Profile Title:

U.S. Government Protection Profile Intrusion Detection System – System for Medium Robustness Environments

Common Criteria Version:

This Protection Profile "U.S. Government Protection Profile Intrusion Detection System – System 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 requirements documented in the PP. 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 the international interpretation. CC version 3.1 used the final CC version 2.3 Security Functional Requirements (SFR)s as the new set of version 3.1 SFRs. Minor changes were made to the SFRs that included some deleted SFRs who functions were transferred to Security Assurance Requirements (SAR)s. The version 3.1 SARs were rewritten by the common criteria international community. The NIAP/CCEVS staff developed an assurance equivalence between the version 2.3 and 3.1 SARs. Those assurance equivalent SARs replaced the SARs in the PP. In going through the PP, there may be minor differences between some SFR in the PP and the new version 3.1 SFRs. These minor differences were not modified to ensure the authors intent was left in tact. 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 internet at: http://www.niap-ccevs.org/cc-scheme/pp/.

Comments on this document should be directed to ppcomments@missi.ncsc.mil. The comments should include the title of the document, the page, the section number, and paragraph number, detailed comments and recommendations.

Table of Contents

List of Tables

1 INTRODUCTION TO THE PROTECTION PROFILE

1.1 Protection Profile Identification

1 Title: US Government Protection Profile(PP) Intrusion Detection System(IDS) – System 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 18 June 2007

6 Keywords: Intrusion Detection, Intrusion Detection System, sensing capability, analyzing capability, scanning capability, Medium Robustness Environments

1.2 Overview of the Protection Profile

7 The US Government IDS - System PP for Medium Robustness Environments (IDS System PP) specifies a set of security functional and assurance requirements for Intrusion Detection System products. An IDS monitors an Information Technology (IT) System for activity that may inappropriately affect the IT System. An IT System may range from a computer system to a computer network. An IDS consists of a sensing capability, an analysis capability and an optional but recommended scanning capability. Sensing and scanning capabilities collect information regarding IT System activity and vulnerabilities, which is then analyzed. Sensing is meant to be a passive capability and scanning is an active capability.

8 Analyzing capabilities perform intrusion analysis and further categorization of the collected information. Scanning capabilities are optional for this PP because a base IDS only needs the capability to sense data from the IT environment being monitored and to have the capability to analyze the sensed data. The Security Target (ST) author is responsible for defining what components comprise the system. One or more components can provide the set of capabilities that are described in this document.

9 IDS System PP-conformant products support the ability to monitor, analyze, and manage a set of IT system resources in order to identify events that may be indicative of potential vulnerabilities in or misuse of those IT resources. IDS System PP-conformant products also provide the ability to protect themselves and their associated data from unauthorized access and modification and ensure accountability for each user's actions.

10 The IDS System PP was constructed to provide a target and metric for the development of IDS Systems. This protection profile identifies security functions and assurances that represent the lowest common set of requirements that must be addressed at a Medium Robustness level by a useful IDS System.

11 The assurance requirements were originally based upon Evaluated Assurance Level (EAL) 4. In order to gain the necessary level of assurance for medium robustness environments, extended requirements have been created both to remove ambiguity in as well as to provide greater assurance than that associated with EAL4. The assurance requirements are presented in Section 5.3.

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

13 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;
- Organizational policies that must be addressed by the TOE;
- Security objectives of the TOE and its environment;
- Functional and assurance requirements to meet those security objectives; and
- Rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats and policies.

14 The IDS System PP is applicable to products regardless of whether they are self-contained, or distributed. In addition, it addresses only security requirements and not any special considerations of any particular product design.

1.3 Conventions

15 Except for replacing United Kingdom spelling with American spelling, the notation, formatting, and conventions used in this PP are consistent with version 2.2 of the CC. Selected presentation choices are discussed here to aid the PP reader.

16 The CC allows several operations to be performed on functional requirements; *refinement*, *selection,* and *assignment* to be added to Part 1 of the CC. Each of these operations is used in this PP.

17 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** and by the '**Refinement:'** label.

18 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.

19 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:].

20 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).

21 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). Applicable CCIMB interpretations are also included in this PP. These will be denoted within the requirement text as an "*Interp Note:*"

22 The CC paradigm also allows protection profile and security target authors to create their own requirements. 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.

23 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

24 See Appendix B for the Glossary.

1.5 Document Organization

25 Section 1, Introduction to the Protection Profile, provides the document management and overview information necessary to identify the PP.

26 Section 2, TOE Description, defines the TOE and establishes the context of the TOE by referencing generalized security functions.

27 Section 3, Security Environment, 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.

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

29 Section 5, IT Security Requirements, specifies the security functional and assurance requirements that must be satisfied by the TOE and the IT environment.

30 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 Strength of Function (SOF) and use of the extended requirement.

31 Section 7, Appendices, includes the appendices that accompany the PP and provides clarity and/or explanation for the reader.

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

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

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

35 Appendix D, Robustness Environment Characterization, 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.

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

2 TOE DESCRIPTION

37 This PP specifies the minimum set of security requirements to satisfy Medium Robustness Environments for a TOE that is an IDS System.

38 Experience has shown that many security compromises occur when products are "composed"; that is, individual products that may be, by themselves, trustworthy, yield a vulnerable result when they are integrated together as a composite product. In order to provide the assurance necessary for products to be integrated into medium robustness environments, it is generally necessary to require that certain components of a product be evaluated as part of a TOE to give high confidence that the product is tamperproof and that the security policy is always invoked (as opposed to allowing an evaluation sponsor to remove the component from the TOE and relegate it to the environment). A particular component of note for all medium robustness products is the product's hardware.

39 Because it is important for medium robustness products to show, through analysis and testing of an evaluation, that they are truly tamperproof and always invoke the correct policy, a medium robustness product's hardware should almost always be specified as part of the TOE that is to be compliant to a medium robustness PP. This is done through the inclusion of ADV_ARC.1 as a requirement for the TOE. In a medium robustness TOE, this requirement cannot be met by the IT Environment, and it is highly unlikely that this requirement can be met without including the underlying hardware (that supports the security functionality provided by the software components of the TOE).

40 It should be noted that inclusion of the hardware within the TOE boundary does not mean that the evidence about this hardware must necessarily be to the same degree of detail as the other portions of the TOE. The level of detail of design documentation and the implementation representation is dependent upon a component's role in security policy enforcement (this applies to software components as well). There must be enough information provided for the hardware and its interaction with the TOE's software to determine the security relevance of the hardware (e.g., does it simply have to work correctly, does it have the ability to bypass policy enforcement, what is the untrusted user interface).

41 The above being said, an IDS claiming conformance to this PP cannot be host based. Medium Robustness assumes that no general purpose computing applications will reside on the TOE.

2.1 Product Type

42 IDS System PP-conformant products support the ability to monitor, analyze, and/or scan a set of IT System resources in order to identify events that may be indicative of potential vulnerabilities in, or misuse of, those IT resources. IDS System PP-conformant products also provide the ability to protect themselves and their associated data from unauthorized access and modification and ensure accountability for each user's actions.

43 The IDS System PP was constructed to provide a target and metric for the development of IDS Systems. This Protection Profile identifies security functions and assurances that represent the minimum set of security requirements that should be addressed at a Medium Robustness level by an IDS System.

44 The IDS System PP is applicable to products regardless of whether they are self-contained or distributed. In addition, it addresses only security requirements and not any special considerations of any particular product design.

2.2 TOE Definition

45 An IDS system has at least two capabilities; a sensing capability and an analysis capability. A scanning capability is recommended but optional. This PP specifies requirements for all three capabilities of an IDS system, including the scanner capability, sensing capability and the analysis capability.

2.3 General TOE Functionality

46 Within the TOE, there are two types of audit data. Audit data related to the system itself is called audit data, and audit data collected by the sensing or scanning capabilities is referred to as IDS audit data. IDS data refers to all TOE Security Functions (TSF) data dealing with the functionality of the IDS (e.g., IDS audit data, signatures, policies, etc.). There are separate administrative roles to manage the different types of TSF data.

47 The IDS System performs the sensing capability, scanning capability and analyzing capability functions.

48 The sensing capability collects information about events (e.g., login, file access, network packets) as they occur. Although real-time analysis would be optimal, it may actually report the events at a later point in time. A sensing capability collects information indicative of inappropriate activity that may have resulted from misuse, improper access, or malicious activity directed at the IT System (i.e., a computer or a network) that the IDS is protecting. The information collected may be obtained from a variety of sources located on an IT System.

49 A scanning capability, if present, collects static configuration information that might be indicative of the potential for a future intrusion or the occurrence of a past

intrusion of an IT System. The information collected may be obtained from a variety of sources located on an IT System. The scanning capability is expected to collect and manage relevant information until it can be delivered to analyses functions.

50 The analyzing capability is expected to receive relevant information from the sensing capability and scanning capability, if present. After the analyzing capability receives the information it will perform a defined set of analyses and respond accordingly. In addition to receiving and analyzing information, an analyzing capability is also expected to protect itself to ensure continuity and integrity of its analyzing capability functions. Some protection, such as physical access, is assumed. Other protection mechanisms, such as the ability to authenticate authorized users and restrict access to functions and data based on administrative roles, must be integrated into the analyzing capability. All management functions related to supporting the security functions of the analyzing capability are included in this PP.

51 An IDS System that is compliant with the IDS System PP provides the following security functions in its evaluated configuration:

- Audit – Section 5.1.1 “Security Audit” describes the TOE’s generation of auditable events, audit records, alarms and audit management. Table 7 lists the minimum set of auditable events. Each auditable event must generate an audit record. 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). The message will be displayed at the various consoles until administrator acknowledgement of the message has occurred. As mentioned in the “Administration” section below, 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 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.

- IDS Audit – The TOE will generate an IDS audit log that contains events about an IT system. These events may include: static configuration information, misuse information, identification and authentication events, service requests, and events based on network traffic. The TOE will then perform analysis based on the information it has collected and generate alarms for potential intrusions. These alarms must be acknowledged by the IDS Administrator. The IDS Administrator must also manage the IDS specific functions including, but not limited to, what data is collected and what analyses will be performed. The TOE must ensure that storage of the IDS audit log is handled in such a way that no data will be lost.

- Encryption – Cryptographic algorithms and key management functions that meet published standards are required in IDS System PP-complaint products. The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. 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. 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 a combination of both. The TOE must perform data encryption/decryption using the Advanced Encryption Standard (AES) with a minimum key size of 128 bits. Additional requirements for key destruction, cryptographic signature, cryptographic operations availability, key agreement, random number generation and cryptographic hashing are provided in section 5.1.2.

- Trusted Channel/ Trusted Path – The TOE is required to provide encrypted communications via a trusted path. Trusted path refers to the encrypted connection used during remote administrative sessions with the TOE.

- Identification and Authentication – The TOE requires multiple Identification and Authentication (I&A) mechanisms for access to services residing on the TOE or for services mediated by the TOE. The type of authentication mechanism required depends on the origin of the source (i.e., remote user or local user from the TOE console) requesting the service.

- Administration – “Administrators” refers to the roles assigned to the individuals responsible for the installation, configuration, and maintenance of the TOE. The TOE requires four separate administrative roles: Cryptographic Administrator, Audit Administrator, IDS 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 and management 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 three administrative roles. The IDS Administrator is solely responsible for regular review of the IDS audit data. The IDS Administrator is also in charge of managing all IDS data. It is important to note that while this PP requires the four administrative roles outlined above, it provides the ST author the option of including additional administrative roles.

2.4 TOE Operation Environment

52 In the case of the IDS System PP, the IT environment must provide a trusted path for remote administrators of the TOE.

3 SECURITY ENVIRONMENT

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

54 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).

55 The remainder of this section addresses the following:

- Threats to TOE assets or to the TOE environment which must be countered;
- Organizational Security Policies;
- Assumptions about the security aspects of a compliant TOE environment.

56 In regards to this PP, the TOE assets are considered to be the TOE security functions and supporting TSF data – in particular IDS audit data.

3.1 Threats

3.1.1 Threat Agent Characterization

57 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

58 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.

59 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.

60 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.*

61 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).

62 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.

63 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.

64 The important general points are:

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

65 Additional explanation about how a Medium Robustness Environment is characterized can be found in Appendix D of this document.

66 The following threats are addressed by the TOE and should be read in conjunction with the threat rationale, Section 6.1. 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.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

Threat Name	Threat Definition
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records and IDS audit records to be lost or modified, or prevent future audit records and IDS 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 compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.EAVESDROP	A malicious user or process may observe or modify user or TSF data transmitted between physically separated parts of the TOE.
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 malicious user, process or external IT entity may masquerade as an authorized entity in order to gain access to data or TOE resources.

Threat Name	Threat Definition
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 (e.g., captured as it was transmitted during the course of legitimate use).
T.RESIDUAL_DATA	A user or a process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.SPOOFING	A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T. UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
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.UNIDENTIFIED_INTRUSIONS	The IDS Administrator may fail to notice potential intrusions, thus limiting the IDS Administrator's ability to identify and take action against a possible intrusion.
T.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.

Threat Name	Threat Definition
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

3.2 Organizational Security Policies

67 An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs.

Table 2 Medium Robustness Applicable Policies

Policy Name	Policy Definition
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 TOE.
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.COMPONENT_IDENTITY	The IDS Administrator will give each TOE component that provides a scanning, sensing, or analyzing capability a unique component Identification (ID).
P.CRYPTOGRAPHIC_FUNCTIONS	The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.

Policy Name	Policy Definition
P.CRYPTOGRAPHY_VALIDATED	Where the TOE requires FIPS-approved security functions, only National Institute of Standards Technology Federal Information Processing Standard Publication (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.IDS_DATA_COLLECTION	IDS audit events based on data collected from IT System resources will be created.
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

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

Table 3 Medium Robustness Applicable Assumptions

Assumption Name	Assumption Definition
A.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.
A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

4 SECURITY OBJECTIVES

69 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

Table 4 Medium Robustness Security Objectives

Objective Name	Objective 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 (i.e., audit records and IDS audit records).
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.

Objective Name	Objective Definition
O.CRYPTOGRAPHY_VALIDATED	The TOE shall use National Institute of Standards Technology- Federal Information Processing Standard (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.
O.IDENTIFIED_COMPONENT	Each component will have a unique component ID assigned by the IDS Administrator.
O.IDS_AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events from targeted IT System resource(s) and associate those events with component that created the record.
O.IDS_AUDIT_REVIEW	The TOE will provide the capability to selectively view IDS audit information, and alert the IDS Administrator of potential intrusions.
O.MAINT_MODE	The TOE shall provide a mode from which recovery or initial start-up procedures can be performed.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.

Objective Name	Objective Definition
O.PROTECT_IN_TRANSIT	The TSF shall protect user and TSF data when it is in transit from one portion of a distributed TOE to another.
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.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
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 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_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.

Objective Name	Objective Definition
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 that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.
O.USER_GUIDANCE	The TOE will provide users with the information necessary to correctly use the security mechanisms.
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

Table 5 Medium Robustness Environmental Security Objectives

Environmental Objective Name	Environmental Objective Definition
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.

Environmental Objective Name	Environmental Objective Definition
OE.MANAGEMENT	The environment will provide a secure communication path with the TSF for the purpose of remote administration of the TOE by authorized administrators.

5 IT SECURITY REQUIREMENTS

5.1 TOE Security Functional Requirements

70 This section defines the functional requirements for the TOE. Functional requirements in this PP were drawn directly 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.

Table 6 Security Functional Requirements

Functional Components (from CC Part 2)	
FAU_ARP.1(1)	Security alarms (Security Violation Alarm)
FAU_ARP.1(2)	Security alarms (IDS Intrusion Alarms)
FAU_ARP_ACK_(EXT).1	Security alarm acknowledgement
FAU_ARP_ACK_(EXT).2	Intrusion alarm acknowledgement
FAU_GEN.1-NIAP-0407	Audit data generation (Audit Records)
FAU_GEN_(EXT).1	Audit data generation (IDS Audit Records)
FAU_GEN_(EXT).3	Audit data generation (Scanning capability)
FAU_GEN.2-NIAP-0410	User identity association (Human Users)
FAU_GEN_(EXT).2	User identity association (IDS Components)
FAU_SAA.1-NIAP-0407	Potential violation analysis
FAU_SAA_(EXT).1	Analyzing capability Intrusion Analysis
FAU_SAR.1(1)	Audit review (Audit Records)
FAU_SAR.1(2)	Audit review (IDS Audit Records)
FAU_SAR.2(1)	Restricted audit review (Audit Records)
FAU_SAR.2(2)	Restricted audit review (IDS Audit Records)
FAU_SAR.3(1)	Selectable audit review (Audit Records)

Functional Components (from CC Part 2)	
FAU_SAR.3(2)	Selectable audit review (IDS Audit Records)
FAU_SEL.1-NIAP-0407(1)	Selective Audit (Audit Events)
FAU_SEL.1-NIAP-0407(2)	Selective Audit (IDS Audit Events)
FAU_STG.1-NIAP-0429	Protected audit trail storage
FAU_STG.2-NIAP-0429	Guarantees of audit data availability
FAU_STG.NIAP-0414-1-NIAP-0429(1)	Site configurable prevention of audit data loss (Audit Records)
FAU_STG.NIAP-0414-1-NIAP-0429(2)	Site configurable prevention of audit data loss (IDS Audit Records)
FCS_BCM_(EXT).1	Baseline cryptographic module
FCS_CKM.1(1)	Cryptographic key generation (for Symmetric Keys)
FCS_CKM.1(2)	Cryptographic key generation (for Asymmetric Keys)
FCS_CKM.2	Cryptographic key distribution
FCS_CKM.4	Cryptographic key destruction
FCS_CKM_(EXT).2	Cryptographic key handling and storage
FCS_COP.1(1)	Cryptographic operation (Encryption/Decryption)
FCS_COP.1(2)	Cryptographic operation (Cryptographic Signatures)
FCS_COP.1(3)	Cryptographic operation (Cryptographic Hashing)
FCS_COP.1(4)	Cryptographic operation (Cryptographic Key Agreement)
FCS_COP_(EXT).1	Random Number Generator
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication failure handling

Functional Components (from CC Part 2)	
FIA_ATD.1(1)	User attribute definition (Human User Identity)
FIA_ATD.1(2)	User attribute definition (Components)
FIA_UAU.2	User authentication before any action
FIA_UID.2(1)	User identification before any action (Human User Identification)
FIA_UID.2(2)	User identification before any action (Component Identification)
FIA_USB.1(1)	User-Subject Binding (Human User-Subject Binding)
FIA_USB.1(2)	User-Subject Binding (Component-Subject Binding)
FMT_MOF.1(1)	Management of security functions behavior (TSF Non-Cryptographic Self Tests)
FMT_MOF.1(2)	Management of security functions behavior (Cryptographic Self Tests)
FMT_MOF.1(3)	Management of security functions behavior (Audit Review)
FMT_MOF.1(4)	Management of security functions behavior (Audit Selection)
FMT_MOF.1(5)	Management of security functions behavior (Security Alarms)
FMT_MOF.1(6)	Management of security functions behavior (IDS Audit Review)
FMT_MOF.1(7)	Management of security functions behavior (IDS Audit Selection)
FMT_MOF.1(8)	Management of security functions behavior (IDS Intrusion Alarms)
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attributes initialization
FMT_MTD.1(1)	Management of TSF data (Cryptographic TSF Data)

Functional Components (from CC Part 2)	
FMT_MTD.1(2)	Management of TSF data (Non-Cryptographic, Non-Time TSF data)
FMT_MTD.1(3)	Management of TSF data (Time TSF Data)
FMT_REV.1	Revocation
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_ITT.1	Basic Internal TSF Data Transfer Protection
FPT_RCV.2	Automated Recovery
FPT_RPL.1	Replay detection
FPT_STM.1	Reliable time stamps
FPT_TST_(EXT).1	TSF testing
FPT_TST.1(1)	TSF testing (Cryptographic)
FPT_TST.1(2)	TSF testing (key generation)
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_TRP.1(1)	Trusted path (Prevention of Disclosure)
FTP_TRP.1(2)	Trusted path (Detection of Modification)

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_ARP.1(1) Security alarms (Security Violation Alarm)

FAU_ARP.1.1(1) **Refinement**: The TSF shall [immediately generate 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 security Administrative sessions that exist;

c) Remote security Administrative sessions that are initiated before the alarm has been acknowledged;

d) 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.

71 *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. 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, email) are included in the TOE. If other methods are specified, the ST author must provide for them through the FMT requirements.*

5.1.1.2 FAU_ARP.1(2) Security Alarms (IDS Intrusion Alarms)

FAU_ARP.1.1(2) **Refinement**: The TSF shall [immediately generate an alarm message, identifying the potential intrusion, and make accessible the analytical result associated with the IDS auditable event(s) that generated the alarm, at the [assignment: alarm destination] and take [assignment: appropriate actions]] upon detection of a potential **intrusion**.

72 *Application Note: There must be an alarm in addition to the audit record generated by the identification of the potential intrusion, though the ST author should refine the nature of the alarm and define its destination (e.g., IDS Administrator console, IDS audit log). The System may optionally perform other actions when intrusions are detected; these actions should be defined in the ST. A violation in this*

requirement applies to any conclusions reached by the analyzing capability related to past, present, and future intrusions or intrusion potential.

5.1.1.3 FAU_ARP_ACK_(EXT).1 Security Alarm Acknowledgement

FAU_ARP_ACK_(EXT).1.1 The TSF shall display the persistent 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 optional audible alarm will sound until acknowledged by a Security 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 Security Administrator sessions that received the alarm.

73 *Application Note: This extended requirement is necessary since a CC requirement does not exist to ensure a Security Administrator will be aware of the alarm. The intent is to ensure that if a Security Administrator is logged in and not physically at the console or remote workstation the message will remain displayed until they have acknowledged it. 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.*

5.1.1.4 FAU_ARP_ACK_(EXT).2 Intrusion Alarm Acknowledgement

FAU_ARP_ACK_(EXT).2.1 The TSF shall display the alarm message identifying the potential intrusion and make accessible the analytical result associated with the IDS auditable event(s) until it has been acknowledged.

FAU_ARP_ACK_(EXT).2.2 The TSF shall display an acknowledgement message identifying a reference to the potential intrusion, 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 IDS Administrator sessions that received the alarm.

74 *Application Note: This extended requirement is necessary since a CC requirement does not exist to ensure a Security Administrator will be aware of the alarm. The intent is to ensure that if a Security 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 or be otherwise obscured due to other activity taking place (e.g., the Audit Administrator is running an audit report).

5.1.1.5 FAU_GEN.1-NIAP-0407 Audit Data Generation (Audit Records)

FAU_GEN.1.1-NIAP-0407 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 as listed in Table 7**;**

c) [selection: [assignment: events at a basic level of audit introduced by the inclusion of additional Security Functional Requirements (SFR) 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”].

75 *Application Note: For the selection, the ST author should choose one or both of the assignments (as detailed in the following paragraphs), or select “no additional events”.*

76 *Application Note: For the first assignment, the ST author augments the table (or lists explicitly) the audit events associated with the basic level of audit for any SFRs that the ST author includes that are not included in this PP.*

77 *Application Note: 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.*

78 *Application Note: If no additional (CC or extended) SFRs are included, or if additional SFRs are included that do not have “basic” audit associated with them, it is acceptable to assign “no additional events” in this item.*

79 *Application Note: The NIAP-0407 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_GEN.1.*

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of Table 7 below].

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

Table 7 Auditable Events Table

Requirement	Auditable Events	Audit Record Contents
FAU_ARP.1(1)	Actions taken due to imminent security violations.	Identification of what caused the generation of the alarm.
FAU_ARP.1(2)	Actions taken due to imminent security intrusions.	Identification of what caused the generation of the alarm.
FAU_ARP_ACK_(EXT).1	Acknowledgement of alarm.	The identity of the administrator that acknowledged the alarm.
FAU_ARP_ACK_(EXT).2	Acknowledgement of alarm.	The identity of the IDS Administrator that acknowledged the alarm.
FAU_GEN.1-NIAP-0407	None.	
FAU_GEN_(EXT).1	None.	
FAU_GEN_(EXT).3	None.	
FAU_GEN.2-NIAP-0410	None.	
FAU_GEN_(EXT).2	None.	
FAU_SAA.1-NIAP-0407	a) Enabling and disabling of any of the analysis mechanisms; b) Automated responses performed by the tool.	The identity of the Security Administrator performing the function.

Requirement	Auditable Events	Audit Record Contents
FAU_SAA_(EXT).1	a) Enabling and disabling of any of the analysis mechanisms; b) Automated responses performed by the tool.	The identity of the IDS Administrator performing the function.
FAU_SAR.1(1)	Reading of information from the audit records.	The identity of the Administrator performing the function.
FAU_SAR.1(2)	Reading of information from the IDS audit records.	The identity of the IDS Administrator performing the function.
FAU_SAR.2(1)	Unsuccessful attempts to read information from the audit records.	The identity of the administrator performing the function.
FAU_SAR.2(2)	Unsuccessful attempts to read information from the audit records.	The identity of the IDS Administrator performing the function.
FAU_SAR.3(1)	None.	
FAU_SAR.3(2)	None.	
FAU_SEL.1-NIAP-0407(1)	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_SEL.1-NIAP-0407(2)	All modifications to the audit configuration that occur while the audit collection functions are operating.	The identity of the IDS Administrator performing the function.
FAU_STG.1-NIAP-0429	None.	
FAU_STG.2-NIAP-0429	None.	

Requirement	Auditable Events	Audit Record Contents
FAU_STG.NIAP-0414-1-NIAP-0429(1)	Actions taken due to the audit storage failure.	The identity of the Security Administrator performing the function.
FAU_STG.NIAP-0414-1-NIAP-0429(2)	Actions taken due to the IDS audit storage failure.	The identity of the IDS Administrator performing the function.
FCS_BCM_(EXT).1	None.	
FCS_CKM.1(1)	Success and failure of the activity.	The object attribute(s), and object value(s) excluding any sensitive information (e.g., secret or private keys).
FCS_CKM.1(2)	Success and failure of the activity.	The object attribute(s), and object value(s) excluding any sensitive information (e.g., secret or private keys).
FCS_CKM.2	Success and failure of the activity.	The object attribute(s), and object value(s) excluding any sensitive information (e.g., secret or private keys).
FCS_CKM.4	Success and failure of the activity.	The object attribute(s), and object value(s) excluding any sensitive information (e.g., secret or private keys).
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.

Requirement	Auditable Events	Audit Record Contents
FCS_COP.1(2)	Failure of cryptographic operation.	Type of cryptographic operation. Any applicable cryptographic mode(s) of operation, excluding any sensitive information.
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	None.	
FDP_ACC.1	None.	
FDP_ACF.1	None.	
FDP_RIP.2	None.	
FIA_AFL.1	a) The reaching of the threshold for the unsuccessful authentication attempts.	a) Identity of the unsuccessfully authenticated user. b) The actions (e.g., disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal).
FIA_ATD.1(1)	None.	
FIA_ATD.1(2)	None.	

Requirement	Auditable Events	Audit Record Contents
FIA_UAU.2	All use of the authentication mechanism.	Claimed identity of the user using the authentication mechanism.
FIA_UID.2(1)	All use of the user identification mechanism.	Claimed identity of the user using the identification mechanism.
FIA_UID.2(2)	All use of the component identification mechanism.	Claimed identity of the component using the identification mechanism.
FIA_USB.1(1)	Success and failure of binding of user security attributes to a subject (e.g., success and failure to create a subject).	The identity of the user whose attributes are attempting to be bound.
FIA_USB.1(2)	Success and failure of binding of component security attributes to a subject (e.g., success and failure to create a subject).	The identity of the component whose attribute was attempting to be bound.
FMT_MOF.1(1)	All modifications in the behavior of the functions in the TSF.	The identity of the Security Administrator performing the function.
FMT_MOF.1(2)	All modifications in the behavior of the functions in the TSF.	The identity of the Cryptographic 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 Security Administrator performing the function.
FMT_MOF.1(5)	All modifications in the behavior of the functions in the TSF.	The identity of the Security Administrator performing the function.

Requirement	Auditable Events	Audit Record Contents
FMT_MOF.1(6)	All modifications in the behavior of the functions in the TSF.	The identity of the IDS Administrator performing the function.
FMT_MOF.1(7)	All modifications in the behavior of the functions in the TSF.	The identity of the IDS Administrator performing the function.
FMT_MOF.1(8)	All modifications in the behavior of the functions in the TSF.	The identity of the IDS Administrator performing the function.
FMT_MSA.1	All modifications of the values of security attributes.	The identity of the Security Administrator performing the function.
FMT_MSA.3	a) Modifications of the default setting of permissive or restrictive rules, b) All modifications of the initial values of security attributes.	The identity of the Security Administrator performing the function.
FMT_MTD.1(1)	All modifications to the values of TSF data.	The identity of the administrator performing the function.
FMT_MTD.1(2)	All modifications to the values of TSF data.	The identity of the administrator performing the function.
FMT_MTD.1(3)	All modifications to the values of TSF data.	The identity of the administrator performing the function.
FMT_REV.1	All attempts to revoke security attributes.	The identity of the Security Administrator performing the function and the identity of the user whose security attributes are being revoked.

Requirement	Auditable Events	Audit Record Contents
FMT_SMF.1	Use of the management functions.	User IDs that are associated with the modifications. The identity of the administrator performing the function.
FMT_SMR.2	a) Modifications to the group of users that are part of a role; b) Unsuccessful attempts to use a role due to the given conditions on the roles.	User IDs that are associated with the modifications. The identity of the administrator performing the function.
FPT_ITT.1	None.	
FPT_RCV.2	a) Failure or service discontinuity; b) Resumption of the regular operation.	Type of failure or service discontinuity.
FPT_RPL.1	Detected replay attacks.	Identity of the user that was the subject of the reply attack.
FPT_STM.1	Changes to the time.	The identity of the administrator who modified the time.
FPT_TST_(EXT).1	Execution of this set of TSF self tests and the results of the tests.	The identity of the administrator performing the test, if initiated by an administrator. Report any results from the test.
FPT_TST.1(1)	Execution of this set of TSF self tests for Cryptography and the results of the tests.	The identity of the administrator performing the test, if initiated by an administrator. Report any results from the test.

Requirement	Auditable Events	Audit Record Contents
FPT_TST.1(2)	Execution of this set of TSF self tests for key generation and the results of the tests.	The identity of the administrator performing the test, if initiated by an administrator. Report any results from the test.
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.
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	Termination of an interactive 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_TRP.1(1)	All attempted uses of the trusted path functions.	Identification of the claimed user identity.

Requirement	Auditable Events	Audit Record Contents
FTP_TRP.1(2)	All attempted uses of the trusted path functions.	Identification of the claimed user identity.

5.1.1.6 FAU_GEN_(EXT).1 Audit Data Generation (IDS Audit Records)

FAU_GEN_(EXT).1.1 The TSF shall be able to generate an IDS audit record by collecting the following information from the targeted IT System resource(s):

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

b) identification and authentication events, service requests, and network traffic;

c) [selection: [assignment: other specifically defined IDS auditable events], "no additional events"].

FAU_GEN_(EXT).1.2 The TSF shall record within each IDS audit record at least the following information:

a) Date and time of the event, type of event**,** component identity.

b) For each IDS audit event type selected in FAU_GEN_(EXT).1.1**,** based on the IDS auditable event definitions of the functional components included in the PP/ST, [information specified in column three of Table 7].

81 *Application Note: The component identity will be a unique identifier given to each component. This will be used to search for IDS audit data created by a particular sensing capability, for example.*

5.1.1.7 FAU_GEN_(EXT).3 Audit Data Generation (Scanning capability)

FAU_GEN_(EXT).3.1 The TSF shall be able to generate an IDS audit record by collecting the following information from the targeted IT System resource(s):

a) [selection: detected malicious code, access control configuration, service configuration, authentication configuration, accountability policy configuration, and detected known vulnerabilities]

b) [selection: [assignment: other specifically defined IDS auditable events], "no additional events"].

FAU_GEN_(EXT).3.2 The TSF shall record within each IDS audit record at least the following information:

a) Date and time of the event, type of event, source/location.

5.1.1.8 FAU_GEN.2-NIAP-0410 User Identity Association (Human Users)

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

82 *Application Note: The NIAP-0410 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_GEN.2.*

5.1.1.9 FAU_GEN_(EXT).2 User Identity Association (IDS Components)

FAU_GEN_(EXT).2.1 For IDS audit events logged by identified scanning/sensing capabilities, the TSF shall be able to associate each auditable event with the identity of the scanning capability and/or sensing capability that logged the event.

5.1.1.10 FAU_SAA.1-NIAP-0407 Potential Violation Analysis

FAU_SAA.1.1-NIAP-0407 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2-NIAP-0407 **Refinement**: The TSF shall **monitor the:**

a) accumulation or combination of the following events known to indicate a potential security violation:

- [Security administrator-specified number of authentication failures;
- Any detected replay of TSF data or security attributes;
- Any failure of the cryptographic self-tests;
- Any failure of the other TSF self-tests;
- Security Administrator-specified number of encryption failures;
- Security Administrator-specified number of decryption failures] known to indicate a potential security violation; and

b) [selection: [assignment: additional events from the set of defined auditable events], “no additional events”]].

83 *Application Note: The intent of this requirement is that an alarm is generated (FAU_ARP.1(1)) once the threshold for one of the events in the bulleted list 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 the bulleted list is intended to be the same value as specified in FIA_AFL.1.1.

84 *Application Note: The failure of TSF self-tests in (a) includes failures of FPT_TST.1(1) and FPT_TST.1(2).*

85 *Application Note: The NIAP-0407 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_SAA.1.*

5.1.1.11 FAU_SAA_(EXT).1 Analyzing Capability Intrusion Analysis

FAU_SAA_(EXT).1.1 The TSF shall perform at least one of the following analysis functions on all IDS audit data received:

a) Statistical analysis which identifies deviations from normal patterns of behavior, and/or

b) Signature analysis which uses patterns corresponding to known attacks or misuses of a System, and/or

c) Integrity analysis which compares System settings or user activity at some point in time with those of another point in time to detect differences; and/or

d) [assignment: other analytical functions]

then create an analytical result for each potential intrusion.

FAU_SAA_(EXT).1.2 The TSF shall create an IDS audit record for each analytical result with at least the following information:

a) Date and time of the result, type of analysis, outcome of analysis, Analyzer component ID, IDS audit records that generated potential intrusion; and

b) [assignment: other security relevant information about the result].

5.1.1.12 FAU_SAR.1(1) Audit Review (Audit Records)

FAU_SAR.1.1(1) The TSF shall provide [the Audit Administrator] with the capability to read [all audit information] from the audit records.

FAU_SAR.1.2(1) The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.1.13 FAU_SAR.1(2) Audit Review (IDS Audit Records)

FAU_SAR.1.1(2) **Refinement**: The TSF shall provide [the IDS Administrator] with the capability to read [all IDS audit information] from the **IDS** audit records.

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

5.1.1.14 FAU_SAR.2(1) Restricted Audit Review (Audit Records)

FAU_SAR.2.1(1) The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.15 FAU_SAR.2(2) Restricted Audit Review (IDS Audit Records)

FAU_SAR.2.1(2) **Refinement**: The TSF shall prohibit all users read access to the **IDS** audit records, except those users that have been granted explicit read-access.

5.1.1.16 FAU_SAR.3(1) Selectable Audit Review (Audit Records)

FAU_SAR.3.1(1) The TSF shall provide the ability to perform *searches and sorting* of audit data based on [date and time, subject identity, type of event, and success or failure of related event].

5.1.1.17 FAU_SAR.3(2) Selectable Audit Review (IDS Audit Records)

FAU_SAR.3.1(1) **Refinement**: The TSF shall provide the ability to perform *searches and sorting* of **IDS** audit data based on [date and time, component identity, type of event].

5.1.1.18 FAU_SEL.1-NIAP-0407(1) Selective Audit (Audit Events)

FAU_SEL.1.1-NIAP-0407(1) **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) [selection: object identity, subject identity, host identity, “none”];

d) [success of auditable security events;

e) Failure of auditable security events; and

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

86 *Application Note: “event type” is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.*

87 *Application Note: The NIAP-0407 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_SEL.1.*

5.1.1.19 FAU_SEL.1-NIAP-0407(2) Selective Audit (IDS Audit Events)

FAU_SEL.1.1-NIAP-0407(2) **Refinement**: The TSF shall **allow only the IDS Administrator** to include or exclude **IDS** auditable events from the set of **IDS** audited events based on the following attributes:

a) *Event typ*e

b) [component identity; and

c) [selection: [assignment: list of additional attributes that **IDS** audit selectivity is based upon], “no additional attributes”].]

5.1.1.20 FAU_STG.1-NIAP-0429 Protected Audit Trail Storage

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

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

88 *Application Note: The NIAP-0429 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_STG.1.*

5.1.1.21 FAU_STG.2-NIAP-0429 Guarantees of Audit Data Availability

FAU_STG.2.1-NIAP-0429 **Refinement**: The TSF shall **restrict the deletion of** stored **IDS** audit records in the **IDS** audit trail to the **IDS Administrator**.

FAU_ STG.2.2-NIAP-0429 **Refinement**: The TSF shall be able to *prevent* unauthorized modifications to the **IDS** audit records in the **IDS** audit trail.

89 *Application Note: Authorized deletion of IDS audit data is not considered a modification of IDS audit data in this context. This requirement applies to the actual content of the IDS audit record, which should be protected from any modifications. The IDS Administrator is allowed to delete the audit records so that would not be considered and unauthorized modification.*

90 *Application Note: The NIAP-0429 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_STG.2.*

FAU_ STG.2.3-NIAP-0429 **Refinement:** The TSF shall ensure that [assignment: metric for saving **IDS** audit records] **IDS** audit records will be maintained when the following conditions occur: [selection: ***IDS audit storage exhaustion, failure, attack***].

91 *Application Note: The ST needs to define the amount of IDS audit data that could be lost under the identified scenarios.*

5.1.1.22 FAU_STG.NIAP-0414-1-NIAP-0429(1) Site-configurable Prevention of Audit Data Loss (Audit Records)

FAU_STG.NIAP-0414-1.1-NIAP-0429(1) **Refinement**: The TSF shall provide the **Audit** Administrator the capability to select one or more of the following actions [selection: 'ignore auditable events', 'prevent auditable events **from being logged**, except those taken by the authorized administrator with special rights', 'overwrite the oldest stored audit records'] and [assignment: other actions to be taken in case of audit store failure] to be taken if the audit trail is full.

FAU_STG.NIAP-0414.1.2-NIAP-0429(1) **Refinement:** The TSF shall [selection: choose one of: 'ignore auditable events', 'prevent auditable events **from being logged**, except those taken by the authorized administrator with special rights', 'overwrite the oldest stored audit records'] and [assignment: other actions to be taken in case of audit storage failure] if the audit trail is full.

92 *Application Note: The TOE provides the audit Administrator the option of preventing audit data loss by preventing auditable events from being logged. The Audit Administrator's actions under these circumstances are not required to be audited.*

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

94 *Application Note: The NIAP-0414-1.1-NIAP-0429 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_STG.4.*

5.1.1.23 FAU_STG.NIAP-0414-1-NIAP-0429(2) Site-configurable Prevention of Audit Data Loss (IDS Audit Records)

FAU_STG.NIAP-0414-1.1-NIAP-0429(2) **Refinement**: The TSF shall provide the **IDS Administrator** the capability to select one or more of the following actions [selection: 'ignore **IDS** auditable events', 'prevent **IDS** auditable events **from being logged**, except those taken by the authorized user with special rights', 'overwrite the oldest stored **IDS** audit records'] and [assignment: other actions to be taken in case of IDS audit store failure] to be taken if the **IDS** audit trail is full.

FAU_STG.NIAP-0414.1.2-NIAP-0429(2) **Refinement**: The TSF shall [selection: choose one of: "ignore **IDS** auditable events", "prevent **IDS** auditable events, except those taken by the authorized user with special rights", "overwrite the oldest stored **IDS** audit records"] and [assignment: other actions to be taken in case of **IDS** audit storage failure] if the **IDS** audit trail is full.

95 *Application Note: The NIAP-0414-NIAP-0429 extension on this requirement is an interpretation of the CC Part 2 requirement FAU_STG.1.*

96 *Application Note: The selections chosen in FAU_STG.NIAP-0414.1.2-NIAP-0429(2) should be identical to the selections chosen for FAU_STG.NIAP-0414.1.1-NIAP-0429(2).*

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 Extended: Baseline Cryptographic Module (FCS_BCM_(EXT))

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

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

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

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

Application Note: This Protection Profile shall use the term "FIPS 140-2" for simplicity.

FIPS PUB 140-2 is currently undergoing a regular five year review; in the near future, FIPS PUB 140-3 will supersede it. Security Targets written to comply with this Protection Profile may replace it with the successor standard that is in force at the time of evaluation.

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

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

(1) Entirely in hardware shall have a minimum overall rating of FIPS PUB 140-2, Level 3,

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

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

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

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

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.

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

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

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

FCS_CKM_(EXT).2.2 The TSF shall store 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 message digest size of *[selection: one or more of 256 bits, 384 bits, 512 bits].***

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

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

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

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

(1) [assignment: Finite Field-based key agreement algorithm]

and cryptographic key sizes (modulus) of [assignment: 2048 bits or greater], or

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

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

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

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

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

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

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

(1) one or more independent hardware-based entropy sources, and/or

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

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

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

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

97 *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.*

5.1.3 User Data Protection (FDP)

5.1.3.1 FDP_ACC.2 Complete Access Control

FDP_ACC.2.1 The TSF shall enforce the [Discretionary Access Control policy] on [all subjects and all named objects] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TOE Scope of Control (TSC) and any object within the TSC are covered by an access control SFP**.**

5.1.3.2 FDP_ACF.1 Security Attribute Based Access Control

FDP_ACF.1.1 **Refinement**: The TSF shall enforce the [Discretionary Access Control policy] to **named** objects based on the following **types of subject and object security attributes**:

a) [the authorized user identity and group membership(s) associated with a subject and

b) the (authorized user (or group) identity, access operations) pairs associated with a named object].

98 *Application Note: This requirement is worded to include only implementations where access control attributes are associated with objects rather than subjects. This implementation becomes critical when satisfying FMT_MTD.1.1(3) and FMT_REV.1.1(1).*

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

- **The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that objects are protected from unauthorized access according to the following ordered rules**:

a) [If the requested mode of access is denied to that authorized user, deny access.

b) If the requested mode of access is permitted to any group of which the authorized user is a member, grant access

c) Else deny access].

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

a) [Authorized administrators must follow the above -stated Discretionary Access Control policy, except after taking the following specific actions: [assignment: list of specific actions].

b) The enforcement mechanism (i.e., access control lists) shall allow authorized users to specify and control sharing of named objects by individual user identities and group identities and shall provide controls to limit propagation of access rights.

c) [assignment: other rules, based on security attributes, that explicitly authorize access of subjects to named objects]].

99 *Application Note: This element allows specifications of additional rules for authorized administrators to bypass the Discretionary Access Control policy for system management or maintenance (e.g., system backup).*

FDP_ACF.1.4 **Refinement**: The TSF shall explicitly deny access of subjects to **named** objects based on the **following rules**:

a) [If the requested mode of access is denied to that authorized user, deny access.

b) If the requested mode of access is denied to every group of which the authorized user is a member, deny access

c) These access controls shall be capable of specifically excluding access to the granularity of a single user].

5.1.3.3 FDP_RIP.2 Full Residual Information Protection

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

5.1.4 Identification and Authentication (FIA)

5.1.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 **Refinement**: The TSF shall detect when *a* **Security** *Administrator configurable positive integer* of unsuccessful authentication attempts occur related to [a user’s authentication *within* [assignment: Security Administrator configurable amount of time].]

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [lock the device for a Security Administrator configurable amount of time].

5.1.4.2 FIA_ATD.1(1) User Attribute Definition (Human User Identity)

FIA_ATD.1.1(1) The TSF shall maintain the following list of security attributes belonging to individual users:

a) [User identity;

b) Authentication data;

c) Authorizations; and

d) [assignment: any other security attributes]].

100 *Application Note: At a minimum, there must be sufficient user information for identification and authentication purposes. That information includes maintaining any authorizations an administrator may possess.*

5.1.4.3 FIA_ATD.1(2) User Attribute Definition (Component Identity)

FIA_ATD.1.1(2) **Refinement**: The TSF shall maintain the following list of security attributes belonging to individual **components**:

a) [Component identity;

b) [assignment: any other security attributes]].

5.1.4.4 FIA_UAU.2 User Authentication Before Any Action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.5 FIA_UID.2(1) User Identification Before Any Action (Human Users)

FIA_UID.2.1(1) The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.6 FIA _UID.2(2) User identification before any action (Components)

FIA_UID.2.1(2) **Refinement**: The TSF shall require each **component** to identify itself before allowing any other TSF-mediated actions on behalf of that **component**.

5.1.4.7 FIA_USB.1(1) User-Subject Binding (Human User-Subject Binding)

FIA_USB.1.1(1) The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [all attributes listed in FIA_ATD.1(1)].

FIA_USB.1.2(1) The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [none].

FIA_USB.1.3(1) The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [only the Security Administrator can change security attributes].

5.1.4.8 FIA_USB.1(2) User-Subject Binding (Component-Subject Binding)

FIA_USB.1.1(2) **Refinement**: The TSF shall associate the following **component** security attributes with subjects acting on the behalf of that **component**: [all attributes listed in FIA_ATD.1(2)].

FIA_USB.1.2(2) **Refinement**: The TSF shall enforce the following rules on the initial association of **component** security attributes with subjects acting on the behalf of **component**: [none].

FIA_USB.1.3(2) **Refinement**: The TSF shall enforce the following rules governing changes to the **component** security attributes associated with subjects acting on the behalf of **component**: [only the IDS Administrator can change **component** security attributes].

5.1.5 Security Management (FMT)

5.1.5.1 FMT_MOF.1(1) Management of Security Functions Behavior (TSF Non-cryptographic Self Tests)

FMT_MOF.1.1(1) The TSF shall restrict the ability to *modify the behavior of* the functions [TSF Self-Test (FPT_TST_(EXT).1)] to [the Security Administrator].

101 *Application Note: "Modify the behavior" refers to specifying the interval at which the test 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 Tests)

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to *enable and disable* the functions [TSF Self-Test (FPT_TST.1(1) and FPT_TST.1(2)] to [the Cryptographic Administrator] **immediately after key generation**.

5.1.5.3 FMT_MOF.1(3) Management of Security Functions Behavior (Audit Review)

FMT_MOF.1.1(3) The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions [Audit Review (FAU_SAR.1(1), FAU_SAR.2(1), and FAU_SAR.3(1))] to [an Administrator].

5.1.5.4 FMT_MOF.1(4) Management of Security Functions Behavior (Audit Selection)

FMT_MOF.1.1(4) The TSF shall restrict the ability to *enable, disable, determine the behavior of, and modify the behavior of, or none* the functions

- [Security Audit Analysis (FAU_SAA.1-NIAP-0407); and
- Security Audit (FAU_SEL.1-NIAP-0407(1))]

to [the Security Administrator].

5.1.5.5 FMT_MOF.1(5) Management of Security Functions behavior (Security Alarms)

FMT_MOF.1.1(5) The TSF shall restrict the ability to *enable and disable* the functions [Security Alarms (FAU_ARP.1(1))] to [the Security Administrator].

102 *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(1) does not lend itself to behavior modification. If the ST author were to include additional functionality in FAU_ARP.1(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 (IDS Audit Review)

FMT_MOF.1.1(6) The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions [IDS Audit Review (FAU_SAR.1(2), FAU_SAR.2(2) and FAU_SAR.3(2))] to [the IDS Administrator].

5.1.5.7 FMT_MOF.1(7) Management of Security Functions Behavior (IDS Audit Selection)

FMT_MOF.1.1(7) The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions [IDS Audit Selection (FAU_SEL.1-NIAP-0407(2))] to [the IDS Administrator].

5.1.5.8 FMT_MOF.1(8) Management of Security Functions Behavior (IDS Intrusion Alarms)

FMT_MOF.1.1(8) The TSF shall restrict the ability to *enable and disable* the functions

- [Analyzing capability Intrusion Analysis (FAU_SAA_(EXT).1); and
- [IDS Intrusion Alarms (FAU_ARP.1(2))]

to [the IDS Administrator].

5.1.5.9 FMT_MSA.1 Management of Security Attributes

FMT_MSA.1.1 The TSF shall enforce the [Discretionary Access Control Policy] to restrict the ability to *change* the security attributes [listed in FDP_ACF.1.1] to [the Security Administrator and owners of the object].

5.1.5.10 FMT_MSA.3 Static Attributes Initialization

FMT_MSA.3.1 The TSF shall enforce the [Discretionary Access Control policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.

103 *Application Note: The TOE must provide protection by default for all objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.*

FMT_MSA.3.2 The TSF shall allow the [Security Administrator] to specify alternative initial values to override the default values when an object or information is created.

5.1.5.11 FMT_MTD.1(1) Management of TSF data (Cryptographic TSF Data)

FMT_MTD.1.1(1) The TSF shall restrict the ability to *modify* the [cryptographic security data] to [the Cryptographic Administrator].

104 *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(2) Management of TSF Data (Non-cryptographic, Non-time TSF data)

FMT_MTD.1.1(2) The TSF shall restrict the ability to [selection: change default, query, modify, delete, clear, [selection: [assignment: other operations], none]] 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].

5.1.5.13 FMT_MTD.1(3) Management of TSF Data (Time TSF Data)

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

105 *Application Note: The ST author is able to restrict the ability to set the time and date to the Security Administrator.*

5.1.5.14 FMT_REV.1 Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the *users* within the TSC to [the Security Administrator].

FMT_REV.1.2 The TSF shall enforce the rules [assignment: specification of revocation rules].

5.1.5.15 FMT_SMF.1 Specification of Management Functions

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

a) [TSF non-cryptographic self tests

b) Cryptographic self tests

c) Audit review

d) Audit selection

e) Security alarms

f) IDS intrusion alarms

g) IDS audit selection

h) IDS audit review].

5.1.5.16 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles

a) [Security Administrator;

b) Audit Administrator;

c) IDS Administrator;

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

e) [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:

a) [All roles shall be able to administer the TOE locally;

b) All roles shall be able to administer the TOE remotely;

c) All roles are distinct; that is, there shall be no overlap of operations performed by each role, with the following exceptions:

   a. All administrators can review the audit trail; and

   b. All administrators can invoke the self-tests]

are satisfied.

106 *Application Note: Only the administrative role has the ability to administer the TOE.*

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 **Refinement**: The TSF shall protect TSF data from *disclosure and modification* **through the use of encryption** when it is transmitted between **physically-**separate**d** parts of the TOE.

107 *Application Note: This requirement will be used for securely transferring data to and from trusted IT entities (e.g., analyzing capability, sensing capability, and scanning capability).*

108 *Application Note: The System IDS PP uses FPT_ITT, FPT_SEP and FPT_RVM to meet the equivalent of FPT_ITA, FPT_ITC and FPT_ITI requirements which are used in the individual component IDS PPs.*

5.1.6.2 FPT_RCV.2 Automated Recovery

FPT_RCV.2.1 When automated recovery from [a failure or service discontinuity] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

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

5.1.6.3 FPT_RPL.1 Replay Detection

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

FPT_RPL.1.2 The TSF shall perform:

a) [reject data;

b) Audit event; and

c) [assignment: list of specific actions]]

when replay is detected.

5.1.6.4 FPT_STM.1 Reliable Time Stamps

109 *Application Note: The following requirement complies with Common Criteria Evaluation & Validation Scheme (CCEVS) Precedent Decision (PD) 107.*

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

5.1.6.5 **Extended: TSF Testing (FPT_TST_(EXT).1)**

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

FPT_TST_(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 .*

5.1.6.6 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 .

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

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

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

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

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

TSF-provided cryptographic functions.v

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

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

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

5.1.7 TOE Access (FTA)

5.1.7.1 FTA_SSL.1 TSF-initiated Session Locking

FTA_SSL.1.1 **Refinement**: The TSF shall lock a **local** interactive session after [a Security Administrator-specified time period of inactivity] by:

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

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

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: [user to re-authenticate].

5.1.7.2 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: [user to re-authenticate].

5.1.7.3 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].

5.1.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 **Refinement**: Before establishing a user session **that requires authentication**, the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.

110 *Application Note: The access banner applies whenever the TOE will provide a prompt for identification and authentication (e.g., 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 Administrators choose, they can remove banner information that informs the user of the product and version number).*

5.1.7.5 FTA_TSE.1 TOE Session Establishment

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

111 *The ST author must define what is meant by "location." For example, it could refer to remote or local sessions or network location.*

5.1.8 Trusted Path/Channels (FTP)

112 *Application Note: Trusted path requirements are only required to be used for identification and authentication, both locally and remotely.*

5.1.8.1 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* users 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) The TSF shall permit *remote users* to initiate communication via the trusted path.

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

113 *Application Note: The encryption used to protect the communication channel from disclosure is either the symmetric algorithm specified in FCS_CKM.1(1) or the asymmetric algorithm specified in FCS_CKM.1(2).*

114 *Application Note: "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.

5.1.8.2 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 *remote* users 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) The TSF shall permit *remote users* to initiate communication via the trusted path.

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

115 *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).*

116 *Application Note: "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 provides a means for detecting the modification of data that flows through the protected communication path.*

5.2 Security Requirements for the IT Environment

117 This Protection Profile provides functional requirements for the IT Environment. The IT environment includes 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 Trusted Path/Channels (FTP)

5.2.1.1 FTP_TRP.1(3) Trusted path

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

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

FTP_TRP.1.3(3) **Refinement**: The **IT Environment** shall require the use of the trusted path for *initial user authentication*, all remote administration actions, [selection: [assignment: other services for which trusted path is required], "none"].

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

119 *This requirement is levied on the IT environment to ensure that the necessary support exists in the IT environment to communicate securely with the TOE. The FCS family of requirements has 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

120 This section defines the assurance requirements for the TOE. Table 8 summarizes the components for medium robustness. The augmented requirements are in bold print. The TOE assurance requirements for this PP do not map to a CC EAL. The assurance requirements are summarized in the Table 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
	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**

Assurance Class	Assurance Components	Assurance Components Description
	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.

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 explain 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.

ADV_INT.3.3C The TSF internals description shall demonstrate that the entire TSF is well-structured and is not overly complex.

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.

5.3.1.4 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: 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 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 interfaces to confirm that the TSF operates as specified.

5.3.5 Class AVA: Vulnerability assessment

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

Dependencies: ADV_FSP.4 Complete Functional Specification
ADV_IMP.1 Implementation of the TSF
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative User guidance

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

Developer action elements:

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

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

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

Content and presentation of evidence elements:

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

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

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

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

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

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

Evaluator action elements:

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

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

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

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

5.3.5.2 AVA_VAN.4 Methodical vulnerability analysis

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

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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

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

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

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

6 RATIONALE

121 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

Table 9 Rationale for TOE Security Objectives

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted 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 (ADO_DEL.2, AGD_PRE.1, AGD_OPE.1) helps 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 IDS specific policies. Likewise, the IDS Administrator can only modify IDS data and not audit data.
	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(2), FMT_MTD.1(3)) also 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 rule-set, 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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records and IDS audit records to be lost or modified, or prevent future audit records and IDS audit records from being recorded, thus masking a user's action.	O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information (i.e., audit records and IDS audit information).	O.AUDIT_PROTECTION (FAU_SAR.2(1), FAU_SAR.2(2), FAU_STG.1-NIAP-0429, FAU_STG.2-NIAP-0429, , FAU_STG.NIAP-0414-1-NIAP-0429(1), FAU_STG.NIAP-0414-1-NIAP-0429(2), FMT_SMF.1) contributes to mitigating this threat by controlling access to both the audit trail and IDS audit trail. All administrators can view the audit log, and only the IDS Administrator can view the IDS audit log. No one is allowed to modify audit records. The Audit Administrator is the only one allowed to delete audit records in the audit trail. The IDS Administrator is the only user allowed to delete audit records from the IDS audit trail.
	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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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) contributes to countering this threat by ensuring that the architecture of the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trails (i.e., audit trail and IDS audit trail). Likewise, ensuring that the functions that protect the audit trails are always invoked is also critical to the mitigation of this threat.
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.	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) is necessary to mitigate this threat by ensuring no TSF data remain in resources allocated to a user. Even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.
	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 (AVD_ARC) contributes to countering this threat by ensuring that the architecture of the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the cryptographic data and executable code.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	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 an analysis that documents the amount of key information that can be leaked via a covert channel. This provides information that identifies how much material could be inappropriately obtained within a specified time period.
T.EAVESDROP A malicious user or process may observe or modify user or TSF data transmitted between physically separated parts of the TOE.	O.PROTECT_IN_TRANSIT The TSF shall protect user and TSF data when it is in transit from one portion of a distributed TOE to another.	O.PROTECT_IN_TRANSIT (FPT_ITT.1) mitigates the threat of eavesdropping by providing basic internal transfer protection for TSF data.
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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	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.5, ADV_TDS.4, ADV_INT.3, ADV_ARC.1) counters this threat, to a degree, by requiring that the TOE be developed using sound engineering principles. By accurately and completely documenting the design of the security mechanisms in the TOE, including a security model, the design of the TOE can be better understood, which increases the chances that design errors will be discovered.
	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/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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 flawed design threat is mitigated. By controlling who has access to the TOE's implementation representation and ensuring that changes to the implementation are analyzed and made in a controlled manner, the threat of intentional or unintentional errors being introduced into the implementation are reduced.
	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_IMP. 1, ADV_TDS.4, ADV_FSP.5, ADV_TDS.1, ADV_INT.3, ADV_ARC.1, 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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	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, and TOE design) will be discovered through testing.
	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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
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.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) is necessary to mitigate this threat by ensuring no TSF data remain in resources allocated to a user. Even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.
	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) 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.
	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(2), FMT_MTD.1(3), FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MOF.1(5), FMT_MOF.1(6), FMT_MOF.1(7), FMT_MOF.1(8), FMT_MSA.1, FMT_MSA.3, FMT_REV.1, FMT_SMF.1) provides the capability to restrict access to the TSF to those that are authorized to use the functions. Satisfaction of this objective (and its associated requirements) prevents unauthorized access to the TSF functions and data through the administrative mechanisms.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	O.DISPLAY_BANNER (FTA_TAB.1) helps mitigate this threat by providing the Security Administrator with the ability to remove product information (e.g., product name, version number, etc) 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.TRUSTED_PATH The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.	O.TRUSTED_PATH (FTP_TRP.1(1), FTP_TRP.1(2)) plays a role in addressing this threat by ensuring that there is a trusted communication path between the TSF and various users (e.g., remote administrators.). This ensures the transmitted data cannot be compromised or disclosed during the duration of the trusted path. The protection offered by this objective is limited to TSF data, including authentication data.
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(1), FIA_UID.2(1), FIA_UAU.2, FTA_TSE.1) mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanisms, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
		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.
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.	O.CORRECT_TSF_ OPERATION The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	O.CORRECT_ TSF_OPERATION (FPT_TST_(EXT).1, FPT_TST.1(1) and FPT_TST1(2)) ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software, including the cryptographic functions) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced. While these testing activities are necessary for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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 demonstrate the TSF satisfies the security functional requirements and that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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) 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 authentication data. Prevention of replay of authentication data will counter the threat that a user will be able to record an authentication session between a trusted entity (administrative user or trusted IT entity) and then replay it to gain access to the TOE, as well as counter the ability of a user to act as another user.
T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the	O.RESIDUAL_INFORMATION (FDP_RIP.2) counters this threat by ensuring that TSF data are not persistent when resources are released by one user/process and allocated to another user/process.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
or process to another.	resource is reallocated.	
T.SPOOFING A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.	O.TRUSTED_PATH The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.	It is possible for an entity other than the TOE (a subject on the TOE, or another IT entity on the network between the TOE and the end user) to provide an environment that may lead a user to mistakenly believe they are interacting with the TOE, thereby fooling the user into divulging identification and authentication information. O.TRUSTED_PATH (FTP_TRP.1(1), FTP_TRP.1(2)) mitigates this threat by ensuring users have the capability to ensure they are communicating with the TOE providing identification and authentication data to the TOE.
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 user's sessions are locked and remote user's sessions are dropped after a Security Administrator-defined time period of inactivity. Locking the local user's session reduces the opportunity of someone gaining unauthorized access to 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/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
T.UNAUTHORIZED_ ACCESS A user may gain access to user data for which they are not authorized according to the TOE security policy.	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE (FDP_ACC.2, FDP_ACF.1) works to mitigate this threat by requiring that objects are protected using access control items. An access control item contains information about who is allowed to access an object, as well as the allowed modes of access. The settings present in the access control item selected in the access control decision process determine whether or not a user is authorized to access the object. It is required that all objects be covered by this policy. Note that O.SELF_PROTECTION (ADV_ARC.1) ensures that this access control mechanism is always invoked, thus ensuring that users cannot bypass the mechanism to access data for which they are not authorized.
	O.USER_GUIDANC E The TOE will provide users with the information necessary to correctly use the security mechanisms.	O.USER_GUIDANCE (AGD_OPE.1) mitigates this threat by providing the user the information necessary to use the security mechanisms that control access to user data in a secure manner. For instance, the method by which the discretionary access control mechanism (FDP_ACC.2, FDP_ACF.1) is configured, and how to apply it to the data the user owns, is described in the user guidance. If this information were not available to the user, the information may be left unprotected, or the user may mis-configure the controls and unintentionally allow unauthorized access to their data.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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(1), FAU_ARP_ACK_(EXT).1, FAU_SAR.1, FAU_SAR.3(1)) helps to mitigate this threat by providing a variety of mechanisms for monitoring the use of the system. The two basic ways audit review is performed is through analysis of the audit trail produced by the audit mechanism, and through the use of an automated analysis and alarm system. For analyzing the audit trail, the TOE requires an Audit Administrator role. This role is restricted to audit record review and the deletion of the audit trail for maintenance purposes (e.g., backup). A search and sort capability provides an efficient mechanism for the Audit Administrator to view pertinent audit information. In addition to the local Audit Administrator role, the TOE also has the capability to export the audit information to an external audit analysis tool for more detailed or composite audit analysis. The TOE's audit analysis mechanism must consist of a minimum set of configurable audit events that could indicate a potential security violation. Thresholds for these events must be configurable by an appropriate administrative role. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g., set number of authentication failures, self-test failures, etc.) and immediately notifies an administrator once an

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
		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 administrator sessions. If a Security Administrator is not currently logged into the TOE, the message is stored and immediately displayed the next time a Security Administrator logs into the TOE. This message is displayed and will remain on the screen until a Security Administrator acknowledges the message. At this point, all administrators that have received the message will receive notification that the alarm has been acknowledged, who acknowledged the alarm, and the time that it was acknowledged. In addition to displaying the potential security violation, the message must contain all audit records that generated the potential security violation. By enforcing the message content and display, this objective provides assurance that an administrator will be notified of a potential security violation.
T.UNIDENTIFIED_INTRUSIONS The IDS Administrator may fail to notice potential intrusions, thus limiting the IDS Administrator's ability	O.IDS_AUDIT_REVIEW The TOE will provide the capability to selectively view IDS audit information, and alert the IDS	O.IDS_AUDIT_REVIEW (FAU_SAA_(EXT).1, FAU_ARP.1(2), FAU_ARP_ACK_(EXT).2, FAU_SAR.1(2), FAU_SAR.3(2)) helps to mitigate this threat by providing a variety of mechanisms for monitoring the targeted system

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
to identify and take action against a possible intrusion.	Administrator of potential intrusions.	resources. The two basic ways IDS audit review is performed is through analysis of the IDS audit trail produced by the IDS audit mechanism, and through the use of an automated analysis and alarm system. For analyzing the audit trail, the TOE requires an IDS Administrator role. This role is restricted to IDS audit record review and the deletion of the IDS audit trail for maintenance (backup) purposes. A search and sort capability provides an efficient mechanism for the IDS Administrator to view pertinent IDS audit information. The TOE's IDS audit analysis mechanism must consist of a minimum set of analyses that could indicate a potential intrusion. The TOE performs the analyses per configuration and immediately notifies the IDS Administrator(s) once an analytical result has occurred that indicates an intrusion. If a potential intrusion has been detected, the TOE displays a message that identifies the potential intrusion to all IDS Administrative consoles. The consoles include the local TOE console and any active remote IDS Administrator sessions. If an IDS Administrator is not currently logged into the TOE, the message is stored and immediately displayed the next time an IDS Administrator logs into the TOE. This message is displayed and will remain on the screen until an IDS Administrator acknowledges the message. At this point, all IDS

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
		Administrators that have received the message will receive notification that the alarm has been acknowledged, who acknowledged the alarm, and the time that it was acknowledged. In addition to displaying the potential intrusion, the message must contain all analytical results that generated the potential intrusion. By enforcing the message content and display, this objective provides assurance that a TOE IDS Administrator will be notified of a potential intrusion.
T.UNKNOWN_STATE When the TOE is initially started or restarted after a failure, 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.MAINT_MODE (FPT_RCV.2) helps to mitigate this threat by ensuring that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. After a failure, the TOE enters a state that disallows operations and requires an administrator to follow documented procedures to return the TOE to a secure state.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	O.CORRECT_ TSF_OPERATION The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.	O.CORRECT_TSF_OPERATION (FPT_TST_(EXT).1, FPT_TST.1(1) and FPT_TST.1(2)) counters this threat by ensuring that the TSF runs a suite of tests to successfully demonstrate the correct operation of the TSF (hardware and software) and the TSF's cryptographic components at initial startup of the TOE. In addition to ensuring that the TOE's security state can be verified, an administrator can verify the integrity of the TSF's data and stored code as well as the TSF's cryptographic data and stored code using the TOE-provided cryptographic mechanisms.
	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_ARC.1) works to mitigate this threat by requiring that the TOE developers provide accurate and complete design documentation of the security mechanisms in the TOE, including a security model. By providing this documentation, the possible secure states of the TOE are described, thus enabling the administrator to return the TOE to one of these states during the recovery process.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	O. ROBUST_ADMIN_ GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O. ROBUST_ADMIN_GUIDANCE (AGD_PRE.1, AGD_OPE.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. This guidance also provides information about the corrective measure necessary when a failure occurs (i.e., how to bring the TOE back into a secure state).
P.ACCESS_BANNER 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 TOE.	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. This is required to be displayed before an interactive administrative session.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
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, FAU_GEN_(EXT).3, FIA_USB.1(1), FAU_SEL.1-NIAP-0407(1)) addresses this policy by providing an audit mechanism to record the actions of a specific user, as well as the capability for a Security Administrator to "pre-select" audit events based on the user ID. The audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). Attributes used in the audit record generation process are also required to be bound to the subject, ensuring users are held accountable.
	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). 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.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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_UID.2(1), FIA_UAU.2) 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.
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 an administrator role to isolate administrative actions.	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.
	O.TRUSTED_PATH The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE.	O.TRUSTED_PATH (FTP_TRP.1(1), FTP_TRP.1(2)) satisfies this policy by requiring that each remote administrative and management session for all trusted users is authenticated and conducted via a secure channel.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	OE.MANAGEMENT	This requirement ensures that the Environment provides the trusted path necessary for administrators to manage the TOE remotely. This requirement protects communicated data (ie, administrative actions) from disclosure and modification.
P.COMPONENT_IDENTITY The IDS Administrator will give each TOE component that provides a scanning, sensing or analyzing capability a unique component Identification (ID).	O.IDENTIFIED_COMPONENT Each component will have a unique component ID assigned by the IDS Administrator.	O.IDENTIFIED_COMPONENT (FIA_ATD.1(2), FIA_UID.2(2), FIA_USB.1(2)) Each component in the specified IDS System will have a unique component Identity that will be assigned by the IDS Administrator. This will allow the IDS Administrator to search IDS audit records based on the component that logged the event, as well as only send IDS audit records to be analyzed that came from a particular component. If there are multiple analyzing capabilities present in the IDS System, this allows Administrators to know which one provided the analysis.
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 for its own use, including encryption/decryption and digital signature operations.	O.CRYPTOGRAPHIC_FUNCTIONS (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)) implements this policy, requiring FIPS-validated cryptographic mechanisms. Functions include symmetric encryption and decryption, digital signatures, as well as key generation and establishment functions.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	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 exchange, and random number generation services).	O.CRYTOGRAPHY_ 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 (FCS_BCM_(EXT).1, FCS_CKM.1(1), FCS_CKM.1(2), FCS_COP.1(3) FCS_COP_(EXT).1) 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_INFO RMATION 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 (FDP_RIP.2) counters this threat by ensuring that TSF data are not persistent when resources are released by one user/process and allocated to another user/process.
P.IDS_DATA_COLLE CTION The TOE will create IDS audit events based on data collected from IT System resources.	O.IDS_AUDIT_GEN ERATION The TOE will provide the capability to detect and create records of security-relevant events from targeted IT System resource(s) and associate those events	O.AUDIT_GENERATION (FAU_GEN_(EXT).1, FAU_GEN_(EXT).2, FAU_GEN_(EXT).3, FIA_USB.1(2), FAU_SEL.1-NIAP-0407(2)) addresses this policy by providing an IDS audit mechanism to create records based on the actions from specific IT System resources, as well as the capability for an IDS

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
	with the component that created the record.	Administrator to “pre-select” IDS audit events based on the component ID. The IDS audit event selection function is configurable during run-time to ensure the TOE is able to capture IDS security-relevant events given changes in threat conditions. Additionally, the IDS Administrator’s ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). Attributes used in the audit record generation process are also required to be bound to the subject, ensuring components are bounded to the IDS audit records they create.
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.	O.VULNERABILIT Y_ANALYSIS_TES T The TOE will undergo appropriate 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.
A.NO_GENERAL_PU RPOSE The administrator ensures there are no	OE.NO_GENERAL_ PURPOSE There will be no general-purpose	The TOE will have no general purpose functionalities available to help ensore secure operation of the TSF.

Threat/Policy/ Assumption	Objectives Addressing the Threat/Policy/ Assumption	Rationale
general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.	computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.	
A.PHYSICAL It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.	The TOE's operating environment provides physical protection of the TOE and its assets.

6.2 Rationale for the Security Objectives and Security Functional Requirements for the Environment

122 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.

6.3 Rationale for TOE Security Requirements

Table 10 Rationale for TOE Security Requirements

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_ROLE The TOE will provide administrator roles to isolate	FMT_SMR.2	FMT_SMR.2 requires that four roles exist for administrative actions: the Security Administrator, who is responsible for configuring most security-relevant

Objective	Requirements Addressing the Objective	Rationale
administrative actions, and to make the administrative functions available locally and remotely.		parameters on the TOE; the Cryptographic Administrator, who is responsible for managing the security data that is critical to the cryptographic operations; the Audit Administrator, who is responsible for reading and deleting the audit trail; and the IDS Administrator who is responsible for all IDS specific functionality and data. 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 theses roles do not overlap, except for running self-tests.
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.1-NIAP-0407 defines the set of events that the TOE must be capable of recording. This requirement ensures that the Audit Administrator has the ability to audit any security relevant events that take 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 amount 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	FAU_GEN.2-NIAP-0410 ensures that the audit records associate a user identity with the auditable event. Although the FIA_ATD.1(1) requirement mandates that a “user ID” be used to represent a user identity, the TOE developer is able to associate different types of user-IDs with different users in order to meet this objective.

Objective	Requirements Addressing the Objective	Rationale
	FAU_GEN_(EXT).3	FAU_GEN_(EXT).3 ensures that the IDS audit records are collected from the targeted IT system resources
	FAU_SEL.1-NIAP-0407(1)	FAU_SEL.1-NIAP-0407(1) allows the Security Administrator to configure which auditable events will be recorded in the audit trail. This provides the Security Administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus, reducing the amount of resources consumed by the audit mechanism and providing the ability to focus on the actions of an individual user. In addition, the requirement has been refined to require that the audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions.
	FIA_USB.1(1)	FIA_USB.1(1) plays a role in 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(5)	FMT_MOF.1(5) restricts the ability to control the behavior of the audit and alarm mechanism to the Security Administrator. The Security Administrator is the only user that controls the behavior of the events that generate alarms and whether the alarm mechanism is enabled or disabled.

Objective	Requirements Addressing the Objective	Rationale
	FAU_SAR.2(1)	FAU_SAR.2(1) restricts the ability to read the audit trail to the administrators, 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).
	FAU_STG.1-NIAP-0429	FAU_STG.1-NIAP-0429 also ensures that no one has the ability to perform unauthorized modifications to the audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the audit trail is maintained.
	FAU_STG.NIAP-0414-1-NIAP-0429(1)	FAU_STG.NIAP-0414-1-NIAP-0429(1) 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 being logged (other than actions taken by the administrator) that would generate an audit record or the audit mechanism will overwrite the oldest audit records with new records.
	FMT_SMF.1	FMT_SMF.1 requires the TOE to provide an Audit Administrator with a facility to backup, recover, and archive audit data ensuring the ability to recover corrupted audit records, and access to a complete history of audit information.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MOF.1(8)	FMT_MOF.1(8) restricts the ability to control the behavior of the IDS audit and alarm mechanism to the IDS Administrator. The IDS Administrator is the only user that controls the behavior of the events that generate alarms and whether the alarm mechanism is enabled or disabled.
	FAU_SAR.2(2)	FAU_SAR.2(2) restricts the ability to read the IDS audit trail to the IDS Administrators, thus, preventing the disclosure of the IDS audit data to any other user. However, the TOE is not expected to prevent the disclosure of IDS audit data if it has been archived or saved in another form (e.g., moved or copied to an ordinary file).
	FAU_STG.2-NIAP-0429	FAU_STG.2-NIAP-0429 restricts the ability to perform authorized deletion of IDS audit records to the IDS Administrator for maintenance purposes. FAU_STG.2-NIAP-0429 also ensures that no one has the ability to modify audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the IDS audit trail is maintained.
	FAU_STG-NIAP-0414-1-NIAP-0429(2)	FAU_STG.NIAP-0414-1-NIAP-0429(2) allows the IDS Administrator to configure the TOE so that if the IDS audit trail does become full, either the TOE will prevent any events from occurring (other than actions taken by the administrator) that would generate an IDS audit record or the IDS audit mechanism will overwrite the oldest IDS audit records with new records.

Objective	Requirements Addressing the Objective	Rationale
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(1)	FAU_ARP.1(1) requires that the alarm be displayed at the local administrative console and at the remote Security Administrative console(s) that exist. For alarms at remote consoles, the alarm is sent either during an established session or upon session establishment (as long as the alarm has not been acknowledged). This is required to increase the likelihood that the alarm will be received as soon as possible. This 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 to allow the administrators to react to the potential security violation without having to search through the audit trail looking for the related events.
	FAU_ARP_ACK_(EXT).1	FAU_ARP_ACK_(EXT).1 requires that an alarm generated by the mechanism that implements the FAU_ARP.1(1) requirement be maintained until an administrator acknowledges it. This ensures that the alarm message will not be obstructed and the administrators will be alerted of a potential security violation. Additionally, this requires that the acknowledgement be transmitted to users that received the alarm, thus ensuring that the set of administrators knows which user, specified in the acknowledgement message, has addressed the alarm.

Objective	Requirements Addressing the Objective	Rationale
	FAU_SAA.1-NIAP-0407	FAU_SAA.1-NIAP-0407 defines the events (or rules) that indicate a potential security violation and will generate an alarm. The triggers for these events are largely configurable by the Security Administrator.
	FAU_SAR.1(1)	FAU_SAR.1(1) is used to provide the Audit Administrator the capability to read all the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the end user to interpret the audit trail. It is expected that the audit information be presented in such a way that the end user can examine an audit record and have the appropriate information (that required by FAU_GEN.2-NIAP-0410) presented together to facilitate the analysis of the audit review. Ensuring the audit data are presented in an interpretable format will enhance the ability of the entity performing the analysis to identify potential security violations.

Objective	Requirements Addressing the Objective	Rationale
	FAU_SAR.3(1)	FAU_SAR.3(1) complements FAU_SAR.1(1) by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3(1) requires the administrators be able to establish the audit review criteria based on a user ID and role so that the actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates and times provides the capability to facilitate the administrator's review of incidents that may have taken place at a certain time. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria.

Objective	Requirements Addressing the Objective	Rationale
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_CMC.4, ALC_CMS.4 contributes to this objective by requiring the developer has 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 and ALC_CMS complements eah other by requiring 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. This aids in understanding how the CM system enforces the control over changes made to the TOE.
	ALC_CMS.4	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.

Objective	Requirements Addressing the Objective	Rationale
	ALC_DVS.1	ALC_DVS.1 requires the developer to 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	ALC_FLR.2 plays a role in satisfying the "analyzed" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws.
	ALC_LCD.1	ALC_LCD.1 requires the developer to document the life-cycle model used in the development and maintenance of the TOE. This life-cycle model describes the procedural aspects regarding the development of the TOE, such as design methods, code or documentation reviews, how changes to the TOE are reviewed and accepted or rejected.

O.CORRECT_TSF_OPERATION
The TOE will provide a 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)
		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

Objective	Requirements Addressing the Objective	Rationale
		provides the end user the ability to discover any failures in the hardware security mechanisms. FPT_TST.1(1) and FPT_TST.1(2) are necessary to ensure the correctness of the TSF software and TSF data. 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 for its own use, including encryption/decryption and digital signature operations.	FCS_CKM.1(1)	These FCS requirements 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 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).

Objective	Requirements Addressing the Objective	Rationale
	FCS_CKM.1(2)	FCS_CKM.1(2) is a requirement that a cryptomodule generate asymmetric keys.
	FCS_CKM.2	FCS_CKM.2 specifies that either a manual, automated, or combination manual and automated key distribution method must be implemented.
	FCS_CKM.4	FCS_CKM.4 specifies the requirements for key zeroization in accordance with NIST-FIPS 140-2.
	FCS_CKM_(EXT).2	FCS_CKM_(EXT).2 provides requirements for key handling and storage. This includes association of keys with the proper entity, error checking, and key lifetimes.
	FCS_COP.1(1)	FCS_COP.1(1) specifies that AES be used to perform encryption and decryption operations.
	FCS_COP.1(2)	FCS_COP.1(2) gives three options for providing the digital signature capability; these requirements also contain requirements for obtaining and generating the domain parameters and key for each of the algorithms

Objective	Requirements Addressing the Objective	Rationale
	FCS_COP.1(4)	Key agreement (FCS_COP.1(4)) 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 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.

Objective	Requirements Addressing the Objective	Rationale
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	This objective deals with the issue of using FIPS 140-2-approved cryptomodules in the TOE. The cryptomodule, as used in the components, must be 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, 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)
	FCS_CKM.1(2)
		FCS_CKM.1(1) and (2) mandate that the cryptomodule must generate keys, and that this key generation must be part of the FIPS-validated cryptomodule.

Objective	Requirements Addressing the Objective	Rationale
	FCS_COP_(EXT).1 FCS_COP.1(3)	FCS_COP_(EXT).1 and FCS_COP.1(3) are similar in that they require that any random number generation and hashing functions, respectively, are part of a FIPS-validated cryptographic module. These requirements do not mandate that the functionality is generally available, but only that it be implemented in a FIPS-validated module if other cryptographic functions need these services.
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 an administrator can establish an interactive session. This banner is under complete control of the Security Administrator.
O.DOCUMENT_KEY_LEAKAGE The bandwidth of channels that can be used to compromise key material shall be documented.	AVA_CCA_(EXT).2	AVA_CCA_(EXT).2 requires that a covert channel analysis be performed on the entire TOE to determine the bandwidth of possible cryptographic key leakage. While there are no requirements to limit the bandwidth, the results of this analysis will provide useful guidance on what the specified lifetime of the cryptographic keys should be in order to reduce the damage due to a key compromise.
O.IDENTIFIED_COMPONENT Each component will have a unique component ID assigned by the IDS Administrator.	FIA_ATD.1(2)	FIA_ATD.1(2) defines the attributes of the components, including a component ID that is used to by the TOE to determine a component's identity. This requirement allows the IDS Administrator to search IDS audit records by a particular component ID.
	FIA_UID.2(2)	FIA_UID.2(2) plays a small role in satisfying this objective by ensuring that every component is identified before the TOE performs any collection of data or any analysis of IDS audit data.

Objective	Requirements Addressing the Objective	Rationale
	FIA_USB.1(2)	FIA_USB.1(2) plays a role in satisfying this objective by requiring a binding of security attributes associated with components that are identified with the subjects that represent them in the TOE.
O.IDS_AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events from targeted IT System resource(s) and associate those events with component that created the record.	FAU_GEN_(EXT).1	FAU_GEN_(EXT).1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the IDS Administrator has the ability to audit any IDS security relevant events that takes place in the targeted IT System resources. This requirement also defines the information that must be contained in the IDS audit record for each auditable event. There is a minimum set of information that must be present in every IDS audit record and this requirement defines that, as well as the additional information that must be recorded for each IDS auditable event.
	FAU_GEN_(EXT).2	FAU_GEN_(EXT).2 ensures that the IDS audit records are associated with a component identity. Although the FIA_ATD.1(2) requirement mandates that a component ID be used, the TOE developer is able to associate different types of component ID's with different components in order to meet this objective.
	FAU_GEN_(EXT).3	FAU_GEN_(EXT).3 ensures that the IDS audit records are collected from the targeted IT system resources

Objective	Requirements Addressing the Objective	Rationale
	FAU_SEL.1-NIAP-0407(2)	FAU_SEL.1-NIAP-0407(2) allows the IDS Administrator to configure which IDS auditable events will be recorded in the IDS audit trail. This provides the IDS 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 IDS audit mechanism and providing the ability to focus on the actions of an individual component. In addition, the requirement has been refined to require that the IDS audit event selection function is configurable during run-time to ensure the TOE is able to capture security-relevant events given changes in threat conditions.
	FIA_USB.1(2)	FIA_USB.1(2) plays a role in satisfying this objective by requiring a binding of security attributes associated with components that are associated with the subjects that represent them in the TOE.
O.IDS_AUDIT_REVIEW The TOE will provide the capability to selectively view IDS audit information, and alert the IDS Administrator of potential intrusions.	FAU_SAA_(EXT).1	FAU_SAA_(EXT).1 defines the analyses that indicate a potential intrusion and will generate an alarm and an analytical result to be created. The triggers for these analyses to occur are largely configurable by the IDS Administrator. Additional analyses may be added by the ST author.

Objective	Requirements Addressing the Objective	Rationale
	FAU_ARP.1(2)	FAU_ARP.1(2) requires that the alarm be displayed at the local IDS Administrative console(s) and at the remote IDS Administrative console(s) when IDS Administrative session(s) exists. For alarms at remote consoles, the alarm is sent either during an established session or upon session establishment (as long as the alarm has not been acknowledged). This is required to increase the likelihood that the alarm will be received as soon as possible. This requirement also dictates the information that must be displayed with the alarm. The potential intrusion is identified in the alarm, as are the analytical results of the events that accumulated and triggered the alarm. The analytical result is necessary, it allows the IDS Administrators to react to the potential intrusion without having to search through the IDS audit trail looking for the what analysis produced the alarm.
	FAU_ARP_ACK_(EXT).2	FAU_ARP_ACK_(EXT).2 requires that an intrusion alarm generated by the mechanism that implements the FAU_ARP requirement be maintained until an IDS Administrator acknowledges it. This ensures that the alarm message will not be obstructed and the IDS Administrators will be alerted of a potential intrusion. Additionally, this requires that the acknowledgement be transmitted to users that received the alarm, thus ensuring that that set of administrators knows that the user specified in the acknowledgement message has addressed the alarm.

Objective	Requirements Addressing the Objective	Rationale
	FAU_SAR.1(2)	FAU_SAR.1(2) is used to provide the IDS Administrator the capability to read all the IDS audit data contained in the IDS audit trail. This requirement also mandates the IDS audit information be presented in a manner that is suitable for the end user to interpret the IDS audit trail. It is expected that the IDS audit information be presented in such a way that the end user can examine an IDS audit record and have the appropriate information presented together to facilitate the analysis of the IDS audit review. Ensuring the IDS audit data are presented in an interpretable format will enhance the ability of the entity performing the analysis to identify potential intrusions.
	FAU_SAR.3(2)	FAU_SAR.3(2) complements FAU_SAR.1(2) by providing the IDS Administrators the flexibility to specify criteria that can be used to search or sort the IDS audit records residing in the IDS audit trail. FAU_SAR.3(2) requires the IDS Administrator be able to establish the IDS audit review criteria based on a component so that the events logged by the component can be readily identified and analyzed. Allowing the IDS Administrators to perform searches or sort the IDS audit records based on dates and times provides the capability to facilitate the IDS Administrator's review of incidents that may have taken place at a certain time. It is important to note that the intent of sorting in this requirement is to allow the IDS Administrators the capability to organize or group the records associated with a given criteria.
O.MAINT_MODE The TOE shall provide a mode from which recovery or	FPT_RCV.2	This objective is met by using the FPT_RCV.2 requirement, which ensures that the TOE does not continue to operate

Objective	Requirements Addressing the Objective	Rationale
initial startup procedures can be performed.		in an insecure state when a hardware or software failure occurs. Upon the failure of the TSF self-tests the TOE will no longer be assured of enforcing its security policies. Therefore, the TOE enters a state that operations cease 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_MTD.1(1)	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_MTD.1(1) 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.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MTD.1(2)	The requirement FMT_MTD.1(2) is intended to be used by the ST author, with possible iterations, to address TSF data that has not already been specified by other FMT requirements. This is necessary because the ST author may add TSF data in assignments that cannot be addressed ahead of time by the PP authors.
	FMT_MTD.1(3)	FMT_MTD.1(3) provides the capability of setting the date and time that is used to generate time stamps to the Security Administrator. It is important to allow this functionality, due to clock drift and other circumstances, but the capability must be restricted.
	FMT_MOF.1(1)	There are several functions in the TSF that need to be enabled or disabled: either in a producer role or a consumer role; the ability to detect attempts to replay operations; and the ability to enable the cryptographic module self-tests to be run after generation of a key. The use of these functions is specified and restricted by the FMT_MOF.1 iterations. FMT_MOF.1(1) allows only the Security Administrator to modify the behaviors of the functions of the TSF self test. This refers specifically to the specification of the time interval at which the test is periodically run, or perhaps selecting a subset of the tests to run.
	FMT_MOF.1(2)	FMT_MOF.1(2) restricts the ability to enable or disable the functions of the Cryptographic self test (immediately after key generation) to the Cryptographic Administrator.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MOF.1(3)	FMT_MOF.1(3) restricts the ability to enable, disable, determine and modify the behavior of the audit function to an administrator. This is not to be confused with configuration of the audit log, which falls under the domain of the Security Administrator only.
	FMT_MOF.1(4)	FMT_MOF.1(4) restricts the ability to enable, disable, determine and modify the behavior of the Security audit Analysis and Security Audit Selection to the Security Administrator.
	FMT_MOF.1(5)	FMT_MOF.1(5) restricts the ability to enable or disable security alarms to the Security Administrator.
	FMT_MOF.1(6)	FMT_MOF.1(6) allows only the IDS Administrator to view the IDS audit log. It also allows the IDS Administrator to search and sort through the IDS audit records based on certain criteria (e.g., time of day, component identifier, type of event).
	FMT_MOF.1(7)	FMT_MOF.1(7) allows the IDS Administrator to set which IDS auditable events are logged. This is done at run-time so the IDS Administrator can change the events based on particular threats.
	FMT_MOF.1(8)	FMT_MOF.1(8) allows the IDS Administrator to perform different analyses or modify which records are used for the analyses in order to detect a potential intrusion. If a potential intrusion is detected an alarm will be displayed and must be acknowledged by the IDS Administrator.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MSA.1	FMT_MSA.1 provides the Security Administrator with the capability to manipulate the security attributes of the subjects and objects in their scope of control that determine the Discretionary Access Control Policy.
	FMT_MSA.3	FMT_MSA.3 requires that, by default, the TOE only allows the owner access to the files until a rule in the ruleset allows it. Only the Security Administrator may override the restrictive default value.
	FMT_REV.1	FMT_REV.1 mitigates this threat by allowing only the Security Administrator the capability to remove a users security attributes. This might be done if a user leaves the company and the account must be deleted, or if a user changes from one administrative role to a different role.
	FMT_SMF.1	The requirement FMT_SMF.1 was introduced as an international interpretation. This requirement specifies functionality that must be provided to administrators of the TOE.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.2	The FDP_ACC.2 and FDP_ACF.1 requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation of access to the user data takes place. FDP_ACC.2 specifies that the subjects under control of the policy are to be defined, and that all operations that involve access to (minimally) the data are controlled by the policy. These objects contain the user data to be protected.

Objective	Requirements Addressing the Objective	Rationale
	FDP_ACF_1	FDP_ACF.1 details the manner in which the user data are to be protected. The basics called for by the requirement is to match a set of attributes associated with a subject to a set of “access control items” associated with the object they wish to access; all applicable Access Control Items (ACIs) need to grant access in order for the subject to perform the operation on the object. The details of how the ACIs are collected and the specific operations supported are specified in the ST, and with the attributes define the security policy to be enforced. Setting the attributes (implementing the security policy) is a function of the Security Administrator.
O.PROTECT_IN_TRANSIT The TSF shall protect TSF data when it is in transit from one portion of a distributed TOE to another.	FPT_ITT.1	FPT_ITT.1 ensures that TSF data is protected while being transmitted between separate portions of the TOE.
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.	FPT_RPL.1	FPT_RPL.1 ensures that replay of authentication data, TSF data, and security attributes will be detected and that, when such an attempt is detected, the TSF will, at least, reject the data and audit the event.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	FCS_CKM.4	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 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.

Objective	Requirements Addressing the Objective	Rationale
	FDP_RIP.2	FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 AGD_PRE.1	ALC_DEL.1 AGD_PRE.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.
	AGD_PRE.1	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 preparative procedures (PRE) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.

Objective	Requirements Addressing the Objective	Rationale
	AGD_OPE.1	The AGD_OPE.1 requirement mandates the developer provide the operational user 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 rule set 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	The AGD_OPE.1 is also intended for non-administrative users, and but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since all users in this TOE are also administrators, this requirement will be trivially satisfied by the administrator guidance.
O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the	FIA_UID.2(1)	FIA_UID.2(1) plays a small role in satisfying this objective by ensuring that every one is identified before the TOE performs any mediated functions.

Objective	Requirements Addressing the Objective	Rationale
TOE and to explicitly deny access to specific users when appropriate.	FIA_AFL.1	FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by remote administrators and authorized IT entities. The requirement enables a Security Administrator a settable threshold that prevents unauthorized users from gaining access to an 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.
	FIA_ATD.1(1)	FIA_ATD.1(1) defines the attributes of users, including a user ID that is used 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 user ID 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.
	FIA_UAU.2	FIA_UAU.2 requires that administrators, authorized IT entities and other users authenticate themselves to the TOE before performing administrative duties.

Objective	Requirements Addressing the Objective	Rationale
	FTA_TSE.1	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 can access the TOE.
	FTA_SSL.1 FTA_SSL.2 FTA_SSL.3	The FTA_SSL family partially satisfies the O. 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 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	ADV_ARC.1 will describe how the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies. The inclusion of ADV_ARC.1 also describes how that the TSF makes policy decisions on all interfaces that perform operations on subjects and objects that are scoped by the policies. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies. This includes controlling the accessibility to interfaces, as well as what access control is provided within the interfaces.

Objective	Requirements Addressing the Objective	Rationale
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.	ADV_FSP.5	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_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. Having a complete understanding of what is available at the TSF interface allows one to analyze this functionality in the context of design flaws.
	ADV_TDS.4	ADV_TDS.4 requires that a design of the TOE be provided. This design describes the architecture of the TOE in terms of subsystems. It identifies which subsystems are responsible for making and enforcing security relevant (e.g., anything relating to an SFR) decisions and provides a description, of how those decisions are made and enforced. Having this level of description helps provide a general understanding of how the TOE works, and may allow the reader to discover flaws in the design.

Objective	Requirements Addressing the Objective	Rationale
	ADV_INT.3	ADV_INT.3 ensures that the design of the TOE has been performed using good software engineering design principles that require a modular design of the TSF. Modular code increases the developer's understanding of the interactions within the TSF, which in turn, potentially reduces the amount of errors in the design. Having a modular design is imperative for evaluator's to gain an appropriate level of understanding of the TOE's design in a relatively short amount of time. The appropriate level of understanding is dictated by other assurance requirements in this PP (e.g., ATE_DPT.2, AVA_CCA_(EXT).2, AVA_VAN.4).
	ADV_ARC.1	ADV_ARC.1 addresses the non-bypassability and domain separation aspects of the TSF, since these need to be analyzed differently from other functional requirements.
	ADV_FSP.5 ADV_TDS.1	ADV_FSP.5 and ADV_TDS.1 is also used to ensure 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.

Objective	Requirements Addressing the Objective	Rationale
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_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.
	ADV_INT.3	When performing the activities associated with the ADV_INT.3 requirement, the evaluators will ensure that the architecture of the implementation is modular and consistent with the architecture presented in the low-level design. Having a modular implementation provides the evaluators with the ability to more easily assess the accuracy of the implementation, with respect to the design. If the implementation is overly complex (e.g., circular dependencies, not well understood coupling, reliance on side-effects) the evaluator may not have the ability to assess the accuracy of the implementation.

Objective	Requirements Addressing the Objective	Rationale
	ADV_TDS.4 ADV_ARC.1	While ADV_TDS.4 ADV_ARC.1 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 low-level design as an aide in understanding the implementation representation. The low-level 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.
	ADV_FSP.5 ADV_TDS.1	ADV_FSP.5 and ADV_TDS.1 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 low-level design is correctly reflected in the implementation representation.
	ALC_TAT.1	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.

Objective	Requirements Addressing the Objective	Rationale
O.THOROUGH_FUNCTIONAL_TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.	ATE_COV.2	In order to satisfy O.THOROUGH_FUNCTIONAL_TESTING, the ATE class of requirements is necessary. 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_FUN.1	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_IND.2	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.

Objective	Requirements Addressing the Objective	Rationale
	ATE_DPT.3	ATE_DPT.3 requires the developer to provide a test coverage analysis that demonstrates depth of coverage of the test suite. This component complements ATE_COV.2 by ensuring that the developer takes into account the high-level and low-level design when developing their test suite. Since exhaustive testing of the TSFI is not required, ATE_DPT.2 ensures that subtleties in TSF behavior that are not readily apparent in the functional specification are addressed in the test suite.
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	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)	FMT_MTD.1(3) satisfies the rest of this objective by providing the capability to set the time used for generating time stamps to the Security Administrator.

Objective	Requirements Addressing the Objective	Rationale

O.TRUSTED_PATH
The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE when supplying identification and authentication data.
	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 (FTP_TRP.1(1)) and modification (FTP_TRP.1(2)). 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). This requirement ensures that the TOE can identify the end points and ensures that a malicious user cannot logically insert themselves between the authenticated user and the TOE. 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 authentication functions. Furthermore, the remote administrator's communication path is encrypted during the entire session.

O.USER_GUIDANCE The TOE will provide users with the information necessary to correctly use the security mechanisms.	AGD_OPE.1	The user guidance required by AGD_OPE.1 meets the objective by describing the discretionary access controls available to the user, and how to set the attributes pertaining to the mechanism. This guidance also instructs the user how to log on to the TOE, and how to choose passwords that will not be easily compromised through a brute force attack.
O.VULNERABILITY_ANA	AVA_VAN.4	To maintain consistency with the overall

Objective	Requirements Addressing the Objective	Rationale
LYSIS_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.		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 developer to perform a systematic search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, 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. As with the functional testing, a key element in this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of moderate (or lower) attack potential to violate the TOE's security policies.
OE.MANAGEMENT The environment will provide a secure communication path with the TSF for the purpose of remote administration of the TOE by authorized administrators.	FTP_TRP.1(3)	This requirement ensures that the Environment provides the trusted path necessary for administrators to manage the TOE remotely. This requirement protects communicated data (ie, administrative actions) from disclosure and modification.

6.4 Rationale for Assurance Requirements

123 Section 5.3 was believed to best achieve the goal of addressing circumstances where developers and users require a moderate level of independently assured security in commercial products.

124 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.

6.5 Rationale for Satisfying all Dependencies

125 The IDS System Protection Profile does not satisfy all the requirement dependencies of the Common Criteria. Table 11 lists each functional requirement from the IDS System Protection Profile with a dependency and indicates whether the dependent requirement was included. Table 12 does the same for assurance requirements. For each dependency not met, an explanation is provided why the dependent was not included in the IDS System Protection Profile.

Table 11 Functional Requirement Dependencies

Requirement	Dependency	Satisfied
FAU_ARP.1(1)	FAU_SAA.1-NIAP-0407	Yes
FAU_ARP.1(2)	FAU_SAA.1	This dependency is satisfied by the requirement FAU_SAA_(EXT).1.
FAU_ARP_ACK_(EXT).1	FAU_SAA.1	This dependency is satisfied by the requirement FAU_SAA_(EXT).1.
FAU_ARP_ACK_(EXT).2	FAU_SAA.1	This dependency is satisfied by the requirement FAU_SAA_(EXT).1.
FAU_GEN.1-NIAP-0407	FPT_STM.1	Yes
FAU_GEN_(EXT).1	FPT_STM.1	Yes

Requirement	Dependency	Satisfied
FAU_GEN_(EXT).3	FPT_STM.1	Yes
FAU_GEN.2-NIAP-0410	FAU_GEN.1-NIAP-0407 FIA_UID.1(1)[1]	Yes
FAU_GEN_(EXT).2	FAU_GEN_(EXT).1 FIA_UID.1(2)[2]	Yes
FAU_SAA.1-NIAP-0407	FAU_GEN.1-NIAP-0407	Yes
FAU_SAA_(EXT).1	FAU_GEN_(EXT).1	Yes
FAU_SAR.1(1)	FAU_GEN.1-NIAP-0407	Yes
FAU_SAR.1(2)	FAU_GEN_(EXT).1	Yes
FAU_SAR.2(1)	FAU_SAR.1(1)	Yes
FAU_SAR.2(2)	FAU_SAR.1(2)	Yes
FAU_SAR.3(1)	FAU_SAR.1(1)	Yes
FAU_SAR.3(2)	FAU_SAR.1(2)	Yes
FAU_SEL.1-NIAP-0407(1)	FAU_GEN.1-NIAP-0407 FMT_MTD.1(2)	Yes
FAU_SEL.1-NIAP-0407(2)	FAU_GEN.1-NIAP-0407 FMT_MTD.1(2)	Yes
FAU_STG.1-NIAP-0429	FAU_GEN.1-NIAP-0407	Yes
FAU_STG.2-NIAP-0429	FAU_GEN.1-NIAP-0407	Yes
FAU_STG.NIAP-0414-1-NIAP-0429(1)	FAU_STG.1-NIAP-0429 FMT_MTD.1(2)	Yes
FAU_STG.NIAP-0414-1-NIAP-0429(2)	FAU_STG.1-NIAP-0429 FMT_MTD.1(2)	Yes

[1] The dependency on FIA_UID.1(1) is satisfied by FIA_UID.2(1) because they are hierarchical.
[2] The dependency on FIA_UID.1(2) is satisfied by FIA_UID.2(2) because they are hierarchical.

Requirement	Dependency	Satisfied
FCS_BCM_(EXT).1	None	N/A
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2[3]	Yes FCS_COP.1(1) and (3) satisfy the dependency on FCS_COP.1.
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2[3]	Yes FCS_COP.1(2) and (4) satisfy the dependency on FCS_COP.1.
FCS_CKM.2	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	Yes FCS_CKM.1(1) and FCS_CKM.1(2) satisfy the dependency on FCS_CKM.1.
FCS_CKM.4	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2[3]	Yes FCS_CKM.1(1) and (2) satisfy the dependency on FCS_CKM.1.
FCS_CKM_(EXT).2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	Yes
FCS_COP.1(1)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	Yes FCS_CKM.1(1) satisfies dependency on FCS_CKM.1

[3] The FMT_MSA.2 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.

Requirement	Dependency	Satisfied
FCS_COP.1(2)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	FCS_CKM.1(2) FCS_CKM.1(2) satisfies dependency on FCS_CKM.1
FCS_COP.1(3)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	FCS_CKM.1(1) satisfies dependency on FCS_CKM.1
FCS_COP.1(4)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2[3]	FCS_CKM.1(2) FCS_CKM.1(2) satisfies dependency on FCS_CKM.1
FCS_COP_(EXT).1	None	N/A
FDP_ACC.2	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1[4] FMT_MSA.3	Yes
FDP_RIP.2	None	N/A
FIA_AFL.1	FIA_UAU.1[5]	Yes
FIA_ATD.1(1)	None	N/A
FIA_ATD.1(2)	None	N/A
FIA_UAU.2	FIA_UID.1[1]	Yes
FIA_UID.2(1)	None	N/A
FIA_UID.2(2)	None	N/A
FIA_USB.1(1)	FIA_ATD.1(1)	Yes

[4] The dependency on FDP_ACC.1 is satisfied by FDP_ACC.2 since they are hierarchical.
[5] The dependency on UAU.1 is satisfied by FIA_UAU.2 because they are hierarchical.

Requirement	Dependency	Satisfied
FIA_USB.1(2)	FIA_ATD.1(2)	Yes
FMT_MOF.1(1)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(2)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(3)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(4)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(5)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(6)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(7)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MOF.1(8)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MSA.1	[FDP_ACC.1[7] or FDP_IFC.1 FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1[6]	Yes

[6] The dependency on FMT_SMR.1 is satisfied by FMT_SMR.2 because they are hierarchical.

[7] The dependency on FDP_ACC.1 is satisfies by FDP_ACC.2 because they are hierarchical.

Requirement	Dependency	Satisfied
FMT_MTD.1(1)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MTD.1(2)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_MTD.1(3)	FMT_SMF.1 FMT_SMR.1[6]	Yes
FMT_REV.1	FMT_SMR.1[6]	Yes
FMT_SMF.1	None	N/A
FMT_SMR.2	FIA_UID.1[1]	Yes
FPT_ITT.1	None	N/A
FPT_RCV.2	AGD_OPE.1	Yes
FPT_RPL.1	None	N/A
FPT_STM.1	None	N/A
FPT_TST_(EXT).1	None	N/A
FPT_TST.1(1)	None	N/A
FPT_TST.1(2)	None	N/A
FTA_SSL.1	FIA_UAU.1[1]	Yes
FTA_SSL.2	FIA_UAU.1[1]	Yes
FTA_SSL.3	None	N/A
FTA_TAB.1	None	N/A
FTA_TSE.1	None	N/A
FTP_TRP.1(1)	None	N/A
FTP_TRP.1(2)	None	N/A

Table 12 Assurance Requirement Dependencies

Component	Dependencies	Satisfied
ADV_ARC.1	ADV_FSP.1 ADV_TDS.1	Yes
ADV_FSP.5	ADV_TDS.1	Yes
ADV_IMP.1	ADV_TDS.3 ALC_TAT.1	Yes
ADV_INT.3	ADV_IMP.1 ADV_TDS.3 ALC_TAT.1	Yes
ADV_TDS.4	ADV_FSP.5	Yes
AGD_OPE.1	ADV_FSP.1	Yes
AGD_PRE.1	None	N/A
ALC_CMC.4	ALC_CMS.1 ALC_DVS.1 ALC_LCD.1	Yes
ALC_CMS.4	None	N/A
ALC_DEL.1	None	N/A
ALC_DVS.1	None	N/A
ALC_FLR.2	None	N/A
ALC_LCD.1	None	N/A
ALC_TAT.1	ADV_IMP.1	Yes
ATE_COV.2	ADV_FSP.2 ATE_FUN.1	Yes
ATE_DPT.3	ADV_ARC.1 ADV_TDS.4 ATE_FUN.1	Yes
ATE_FUN.1	ATE_COV.1	Yes
ATE_IND.2	ADV_FSP.2	Yes

Component	Dependencies	Satisfied
	AGD_OPE.1 AGD_PRE.1 ATE_COV.1 ATE_FUN.1	
AVA_CCA_(EXT).1	ADV_FSP.4 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1	Yes
AVA_VAN.4	ADV_ARC.1 ADV_FSP.2 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1	Yes

6.6 Rationale for Extended Requirements

126 Table 13 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 13 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.

Extended Requirement	Identifier	Rationale
		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 Audit Administrator is running an audit report).
FAU_ARP_ACK_(EXT).2	Intrusion alarm acknowledgment (IDS audit data)	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 IDS Administrator is logged in and not physically at the console or remote workstation the message will remain displayed until the IDS Administrators have acknowledged it. The word “persistent” has been added to the requirement to get this point across. “the message will not be scrolled off the screen…” was removed because requirements are not supposed to be in application notes and the PP should not presume an implementation in which messages are displayed or scrolled off the screen.
FAU_GEN_(EXT).1	Audit Data Generation (IDS Audit Records)	This extended requirement is created to capture security functionality specific to the IDS TOE. The CC requires more in FAU_GEN.1 than is

Extended Requirement	Identifier	Rationale
		needed here.
FAU_GEN_(EXT).2	User identity Association (IDS Components)	This extended requirement was created specifically for the IDS TOE because the CC does not provide a way for non-human entities to be associated with auditable events caused by those entities.
FAU_GEN_(EXT).3	Audit Data Generation (Scanning capability)	This extended requirement is created to capture security functionality specific to the IDS TOE. The CC requires more in FAU_GEN.1 than is needed here.
FAU_SAA_(EXT).1	Analyzing capability intrusion analysis	This extended requirement in necessary because the CC does not provide a means to perform analyses and what information must be contained in the analytical result.
FCS_BCM_(EXT).1	Baseline cryptographic module	The CC does not provide a means of specifying a cryptographic module baseline for implementations developed in hardware, in software, or in hardware/software combinations. FCS_BCM_(EXT).1 provides for the specification of the required FIPS certification based on the implementation baseline.
FCS_CKM_(EXT).2	Cryptographic key handling and storage	The CC does not provide components for key handling and storage. Key access and key destruction components do not address

Extended Requirement	Identifier	Rationale
		keys being transferred within the device nor key archiving when key is not in use. FCS_CKM_(EXT).2 addresses internal key transfer and archiving. It also addresses the handling of storage areas where keys reside.
FCS_COP_(EXT).1	Cryptographic operation (for Random Number Generation)	The CC cryptographic operation components are focused on specific algorithm types and operations requiring specific key sizes. The generation of random numbers can be better stated as an extended component. Neither algorithms nor keys are required to generate random numbers. Random number generators can use any combination of software-based or hardware-based inputs as long as the RNG/PRNG design requirements are met and the required RNG/PRNG tests are successful.
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.

Extended Requirement	Identifier	Rationale
AVA_CCA_(EXT).2	Systematic Cryptographic Module Covert Channel Analysis	This extended assurance requirement is deemed necessary to be performed on the entire TOE to determine the bandwidth of possible cryptographic key leakage in order to reduce the damage due to a key compromise.

6.7 Rationale for Not Addressing Consistency Instructions

127 All consistency instructions were followed from the Consistency Instruction Manual for development of U.S. Government PPs for use in Medium Robustness Environments dated March 1, 2004 with the following exceptions:

Table 14 Medium Robustness Threats Not Applicable to the TOE

Threat Name	Threat Definition	Rationale
T.ADMIN_ROGUE	An administrator's intentions may become malicious resulting in TSF data being compromised.	This threat is resolved by P.ACCOUNTABILITY, which requires all users of the TOE to be responsible for their actions on the TOE, whether they are an administrator or just a user with no special privileges.
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (e.g., example of resources that apply to technology) via a resource exhaustion denial of service attack.	This threat is not included in this PP because administrators do not share resources on the TOE.

7 APPENDICES

128 This section contains all the appendices for this PP.

A REFERENCES

[1] *Common Criteria for Information Technology Security Evaluation*, *CCIMB-2004-01-002,* Version 2.2, January 2004.

[1a] Common Criteria for Information Technology Security Evaluation, *CCMB-2006-09, Version 3.1, September 2006.*

[2] *Consistency Instruction Manual for Development of US Government Protection Profiles for Medium Robustness Environments, Release 2.0,* March 1, 2004.

[3] 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.

[3a] Department of Defense Directive 8500.1, "Information Assurance," October 24, 2002

[3b] Department of Defense Instruction 8500.2, "Information Assurance," February 6, 2003

[4] Guidance and Policy for Department of Defense Global Information Grid Information Assurance, September 22, 1999.

[5] Information Assurance Technical Framework, Version 3.1, September 2002.

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

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

[8] NSA Glossary of Terms Used in Security and Intrusion Detection, Greg Stocksdale, NSA Information Systems Security Organization, April 1998.

[9] Federal Information Processing Standard Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001.

[10] Consistency Instruction Manual For development of US Government Protection Profiles For use in Medium Robustness Environments, Release 2.0, March 1, 2004.

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[8] and the disclosure and modification of data.[9]

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

Active **–** (*scanning capability*) **–** to gain understanding of the IT environment through means that illuminate the environment being scanned.

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.

Availability **–** Timely[10], reliable access to IT resources.

[8] Hardware and software.

[9] Stored or communicated.

Component **–** a single scanning capability, sensing capability or analyzing capability, operating within the TOE configuration

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:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a digital authentication code computed from data.

Cryptographic Module **–** 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.

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

[10] According to a defined metric.

***Discretionary Access Control (DAC)* –** A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. Those controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

***Embedded Cryptographic Module* –** On 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 OTE 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.

***Intrusion* –** Any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource.

***Intrusion Detection* –** Pertaining to techniques which attempt to detect intrusion into an IT System by observation of actions, security logs, or audit data. Detection of break-ins or attempts either manually or via software expert systems that operate on logs or other information available on the network.

***Intrusion Detection System (IDS)* –** A combination of one or more sensing capabilities, and one or more analyzing capabilities and an optional but recommended scanning capability that monitor an IT System for activity that may inappropriately affect the IT System's assets and react appropriately.

***Intrusion Detection System Analyzing Capability* –** The components of an IDS that accepts data from sensing capabilities and scanning capabilities and other IT System resources, and then applies analytical processes and information to derive conclusions about intrusions (past, present, or future).

***Intrusion Detection System Data (IDS data)* –** Data collected and produced by the IDS functions. This could include digital signatures, policies, permissions, and IDS audit data.

***Intrusion Detection System Sensing Capability* –** The component of an IDS that collects real-time events that may be indicative of vulnerabilities in or misuse of IT resources.

***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* –** 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 require the same instance of the object.

***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.

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

***Passive* – (*sensing capability)* –** To gain understanding of the IT environment through means that do not effect or impact the environment being sensed.

***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 are used for the enforcement of the TSP.

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

***Subject* –** An entity within the TSC that causes operation 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
API	Application Program Interface
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
CM	Configuration Management
CSP	Critical Security Parameter
DES	Data Encryption Standard
DoD	Department of Defense
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS-PUB 140-2	Federal Information Processing Standard Publication
GIG	Global Information Grid
GUI	Graphical User Interface
ID	Identification
IDS	Intrusion Detection System
IT	Information Technology
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards Technology
NSA	National Security Agency
PP	Protection Profile
RNG	Random Number Generator
RSA	Rivest, Shamir and Aldeman

SFR	Security Functional Requirement
ST	Security Target
TBD	To Be Determined
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Functions
TSFI	TOE Security Function Interface
TSP	TOE Security Policy
TTAP/CCEVS	Trust Technology Assessment Program/ Common Criteria Evaluation Standard Scheme

D ROBUSTNESS ENVIRONMENT CHARACTERIZATION

D.1 General Environmental Characterization

129 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.

130 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).

131 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, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

D.1.1 Value of Resources

132 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.

D.1.2 Authorization of Entities

133 Authorization that entities (users, administrators and 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).

134 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.

135 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.

D.1.3 Selection of Appropriate Robustness Levels

136 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.

137 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.

138 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:

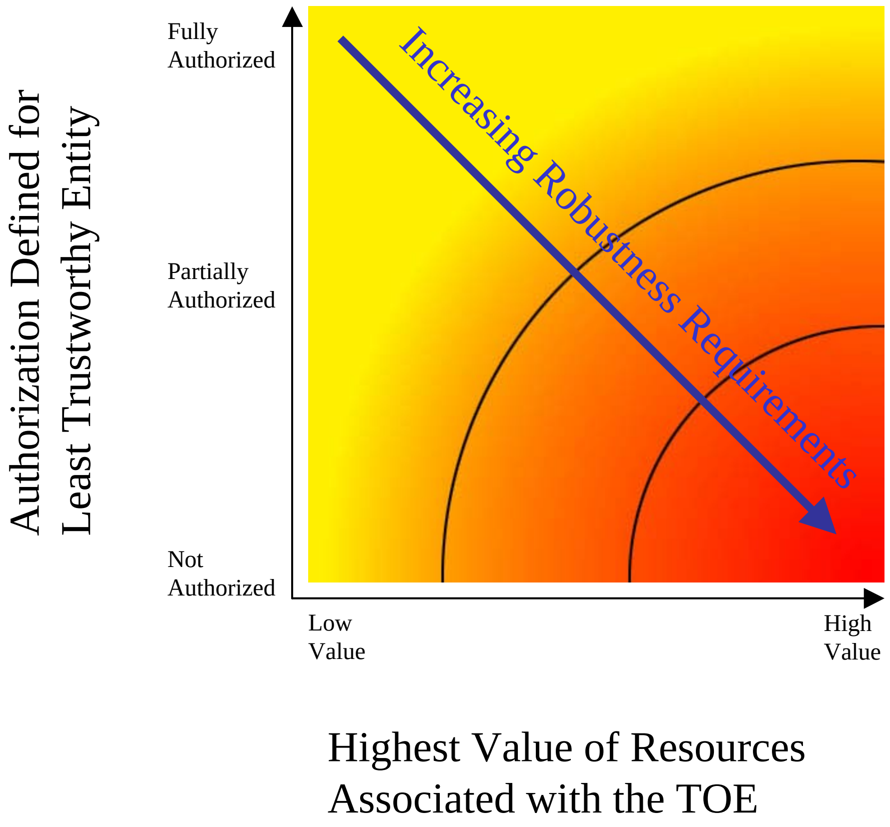
139 The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, 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.

140 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.

141 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.

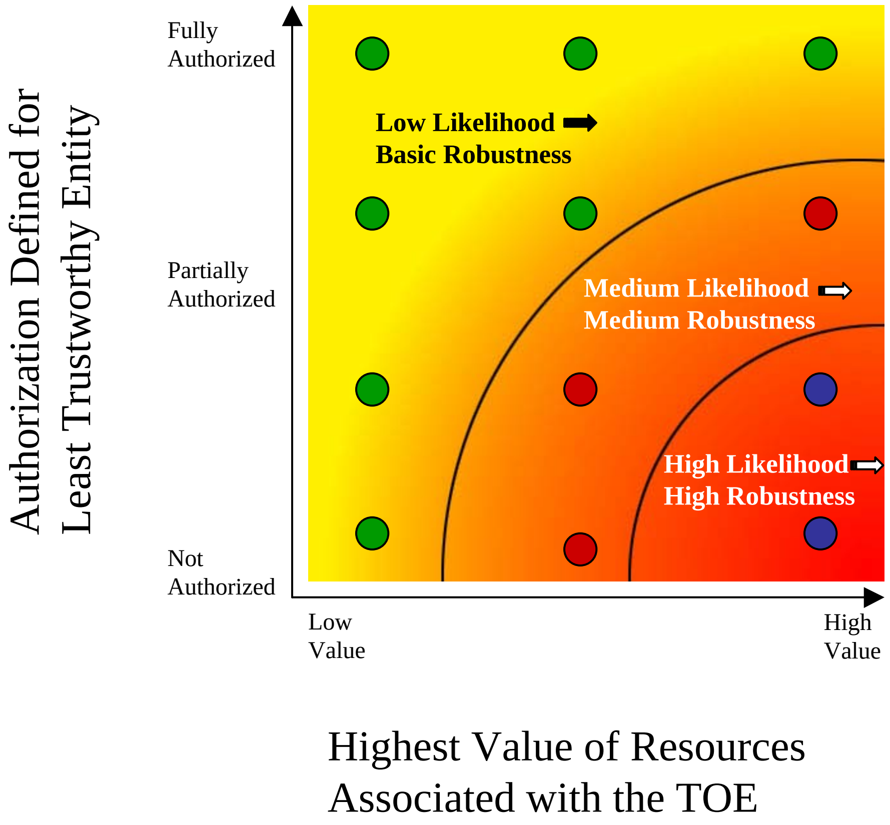
142 As depicted in the following figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. 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.



143 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.

144 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.

145 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 Section 3 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.



146

E REFINEMENTS

147 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.

148 Refinements for the FCS_CKM and FCS_COP SFRs are included as endnotes in this PP. These endnotes are listed immediately following this Appendix.

FAU_ARP.1.1(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:

a) Local console;

b) Remote security Administrative sessions that exist;

c) Remote security Administrative sessions that are initiated before the alarm has been acknowledged;

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.

FAU_ARP.1.1(2) **Refinement**: The TSF shall [immediately generate an alarm message, identifying the potential intrusion, and make accessible the analytical result associated with the IDS auditable event(s) that generated the alarm, at the [assignment: alarm destination] and take [assignment: appropriate actions]] upon detection of a potential (**violation**) **intrusion**.

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 **as listed in Table 7;**

c) [selection: [assignment: events at a basic level of audit introduced by the inclusion of additional Security Functuional Requirements (SFR) 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(2) **Refinement**: For **IDS** audit events (**resulting from actions of**) **logged by** identified (**users**) scanning capabilities and/or sensing capabilities, the TSF shall be able to associate each auditable event with the identity of the (**user**) **scanning capability and/or sensing capability** that (**caused**) **logged** the event.

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

a) accumulation or combination of:

- [Security administrator-specified number of authentication failures;
- Any detected replay of TSF data or security attributes;
- Any failure of the cryptographic self-tests;
- Any failure of the other TSF self-tests;
- Security administrator-specified number of encryption failures;
- Security administrator-specified number of decryption failures] known to indicate a potential security violation:

b) [selection: [assignment: additional events from the set of defined auditable events], “no additional events”]].

FAU_SEL.1.1-NIAP-0407(1) **Refinement**: The TSF shall (**be able to**) **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 typ*e*

c) [selection: object identity, subject identity, host identity, “none”];

d) [success of auditable security events;

e) Failure of auditable security events; and

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

FAU_SEL.1.1-NIAP-0407(2) **Refinement**: The TSF shall (**be able to**) **allow only the IDS Administrator** to include or exclude **IDS** auditable events from the set of **IDS** audited events based on the following attributes:

a) *Event typ*e

b) [component identity;

c) Success of IDS auditable security events;

d) Failure of IDS auditable security events; and

e) [selection: [assignment: list of additional attributes that IDS audit selectivity is based upon], "no additional attributes"].]

FAU_STG.1.1-NIAP-0429 **Refinement**: The TSF shall (**protect the**) **restrict the deletion of** stored audit records (**from unauthorized deletion**) **in the audit trail to the Audit Administrator**.

FAU_STG.2.1-NIAP-0429 **Refinement**: The TSF shall (**protect the**) **restrict the deletion of** stored **IDS** audit records (**from unauthorized deletion**) **in the IDS audit trail to the IDS Administrator**.

FIA_ATD.1.1(2) **Refinement**: The TSF shall maintain the following list of security attributes belonging to individual (**users**) **components**:

a) [Component identity;

b) [assignment: any other security attributes]].

FIA_UID.2.1(2) **Refinement**: The TSF shall require each (**user**) **component** to identify itself before allowing any other TSF-mediated actions on behalf of that (**user**) **component**.

FIA_USB.1.1(2) **Refinement**: The TSF shall associate the following (**user**) **component** security attributes with subjects acting on the behalf of that (**user**) **component**: [all attributes listed in FIA_ATD.1(2)].

FIA_USB.1.2(2) **Refinement**: The TSF shall enforce the following rules on the initial association of (**user**) **component** security attributes with subjects acting on the behalf of (**user**) **component**: [none].

FIA_USB.1.3(2) **Refinement**: The TSF shall enforce the following rules governing changes to the (**user**) **component** security attributes associated with subjects acting on the behalf of (**user**) **component**: [only the IDS Administrator can change (**user**) **component** security attributes].

FTP_TRP.1.1(1) **Refinement**: The TSF shall provide an **encrypted** communication path between itself and *remote* 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.1.1(2) **Refinement**: The TSF shall **use a cryptographic signature to** provide a communication path between itself and *remote* users that is logically distinct from other communication paths and provides assured identification of its end points and (**protection of the channel data from modification or disclosure**) **detection of the modification of data**.

F STATISTICAL RANDOM NUMBER GENERATOR TESTS

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.)

The Monobit Test:

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.

The Poker Test:

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 $0 \leq i \leq 15$.
2. Evaluate the following:

$$X = (16 / 5000) * \left(\sum_{i=0}^{15} [f(i)]^2 \right) - 5000$$

3. The test is passed if 2.16 < X < 46.17.

The Runs Test:

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 ($\geq$ 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**.