

Version 0.71 – May 18, 2006

**Field Device Protection Profile
For SCADA Systems In
Medium Robustness
Environments
Version 0.71**

Prepared For:



**Process Control Security Requirements
Forum (PCSRF)**

Prepared By:

Digital Bond, Inc.

May 18, 2006

Version 0.71 – May 18, 2006

REVISION NOTES

<u>Version</u>	<u>Date</u>	<u>Comments</u>
0.10	Jan 24, 2006	First partial draft with sections 2, 3, and 4
0.50	May 1, 2006	Requirements added
0.70	May 12, 2006	Complete draft for review
		Modified per CCEVS CIM of medium robustness
0.71	May 18, 2006	Minor grammar and spelling changes

TABLE OF CONTENTS

1	INTRODUCTION	5
1.1	PROTECTION PROFILE IDENTIFICATION	5
1.2	PROTECTION PROFILE OVERVIEW	5
1.3	CONVENTIONS.....	6
1.4	DOCUMENT ORGANIZATION.....	7
2	TARGET OF EVALUATION (TOE) DESCRIPTION	8
3	TOE SECURITY ENVIRONMENT.....	10
3.1	MEDIUM ROBUSTNESS	10
3.2	ASSUMPTIONS	11
3.3	THREATS	11
3.4	ORGANIZATIONAL SECURITY POLICIES	16
4	SECURITY OBJECTIVES.....	17
4.1	TOE SECURITY OBJECTIVES	17
4.2	SECURITY OBJECTIVES FOR THE OPERATING ENVIRONMENT	20
5	TOE SECURITY REQUIREMENTS.....	21
5.1	TOE SECURITY FUNCTIONAL REQUIREMENTS	21
5.1.1	<i>Security Functional Components.....</i>	<i>21</i>
5.1.2	<i>Security Audit (FAU) Requirements.....</i>	<i>23</i>
5.1.3	<i>Cryptographic Support (FCS) Requirements</i>	<i>31</i>
5.1.4	<i>User Data Protection (FDP) Requirements</i>	<i>33</i>
5.1.5	<i>Identification And Authentication (FIA) Requirements</i>	<i>37</i>
5.1.6	<i>Security Management (FMT) Requirements.....</i>	<i>39</i>
5.1.7	<i>Protection Of The TSF (FPT) Requirements</i>	<i>46</i>
5.1.8	<i>Resource Utilization (FRU) Requirements</i>	<i>49</i>
5.1.9	<i>TOE Access (FTA) Requirements</i>	<i>50</i>
5.1.10	<i>Trusted Path/Channels (FTP) Requirements.....</i>	<i>51</i>
5.2	ASSURANCE REQUIREMENTS.....	51
5.2.1	<i>Partial CM Automation (ACM_AUT.1).....</i>	<i>53</i>
5.2.2	<i>Generation Support And Acceptance Procedures (ACM_CAP.4).....</i>	<i>53</i>
5.2.3	<i>Problem Tracking CM Coverage (ACM_SCP.2)</i>	<i>54</i>
5.2.4	<i>Detection of Modification (ADO_DEL.2).....</i>	<i>54</i>
5.2.5	<i>Installation, Generation, And Start-Up Procedures (ADO_IGS.1).....</i>	<i>54</i>
5.2.6	<i>Architectural Design (ADV_ARC_(EXP).1).....</i>	<i>55</i>
5.2.7	<i>Functional Specification With Complete Summary (ADV_FSP_(EXP).1).....</i>	<i>55</i>
5.2.8	<i>Security-Enforcing High-Level Design (ADV_HLD_(EXP).1)</i>	<i>56</i>
5.2.9	<i>Modular Decomposition (ADV_INT_(EXP).1).....</i>	<i>57</i>
5.2.10	<i>Subset Of The Implementation Of The TSF (ADV_IMP.1).....</i>	<i>58</i>
5.2.11	<i>Security-Enforcing Low-Level Design (ADV_LLD_(EXP).1)</i>	<i>58</i>
5.2.12	<i>Informal Correspondence Demonstration (ADV_RCR.1)</i>	<i>59</i>
5.2.13	<i>Informal TOE Security Policy Model (ADV_SPM.1)</i>	<i>59</i>
5.2.14	<i>Administrator Guidance (AGD_ADM.1).....</i>	<i>60</i>
5.2.15	<i>User Guidance (AGD_USR.1).....</i>	<i>61</i>
5.2.16	<i>Development Security (ALC_DVS.1).....</i>	<i>61</i>
5.2.17	<i>Flaw Reporting Procedures (ALC_FLR.2).....</i>	<i>61</i>

Version 0.71 – May 18, 2006

5.2.18	<i>Developer Defined Life-Cycle (ALC_LCD.1)</i>	62
5.2.19	<i>Tools and Techniques (ALC_TAT.1)</i>	63
5.2.20	<i>Analysis Of Coverage (ATE_COV.2)</i>	63
5.2.21	<i>Testing: Low-Level Design (ATE_DPT.2)</i>	63
5.2.22	<i>Functional Testing (ATE_FUN.1)</i>	64
5.2.23	<i>Independent Testing-Sample (ATE_IND.2)</i>	64
5.2.24	<i>Systematic Cryptographic Module Covert Channel Analysis (AVA_CCA_(EXP).2)</i>	65
5.2.25	<i>Validation Of Analysis (AVA_MSU.2)</i>	66
5.2.26	<i>Stength Of TOE Security Function Evaluation (AVA_SOF.1)</i>	66
5.2.27	<i>Moderately Resistant (AVA_VLA.3)</i>	66
6	RATIONALE	68
6.1	RATIONALE FOR TOE SECURITY OBJECTIVES	68
6.2	RATIONALE FOR THE SECURITY OBJECTIVES AND SECURITY REQUIREMENTS FOR THE TOE	84
6.3	RATIONALE FOR ASSURANCE REQUIREMENTS	102
6.4	RATIONALE FOR STRENGTH OF FUNCTION CLAIM	102
	APPENDIX A: REFERENCES	104
	APPENDIX B: GLOSSARY	105
	APPENDIX C: ACRONYMS	111
	APPENDIX D: ROBUSTNESS ENVIRONMENT CHARACTERIZATION	112

1 Introduction

This Protection Profile is sponsored by the Process Control Security Requirements Forum (PCSRF) and is intended for the following uses:

1. For vendors, this Protection Profile defines the requirements, as identified by the SCADA community participating in PCSRF, which must be addressed by SCADA field devices such as PLC's, RTU's and IED's in a vendor's Security Target.
2. For SCADA asset owners, this Protection Profile is useful in identifying requirements that can be considered in purchasing specifications. Alternately, asset owners can require products to demonstrate compliance with this Protection Profile.

Any Security Target claiming compliance to this Protection Profile must do so in a strict manner.

1.1 Protection Profile Identification

Title: Field Device Protection Profile for SCADA Systems in Medium Robustness Environments

Sponsor: Process Control Security Requirements Forum (PCSRF)

Author: Digital Bond, Inc.

CC Version: Common Criteria (CC) Version 2.3 and applicable NIAP interpretations from the Consistency Instruction Manual for Medium Robustness Environments dated 1 February 2005

Registration: <to be provided upon registration>

Protection Profile Version: Version 0.7, dated 12 May 2006

Keywords: SCADA, DCS, PLC, RTU, IED, Field Device, Field Controller

1.2 Protection Profile Overview

This Protection Profile specifies the minimum security requirements for SCADA field devices used by a U.S. Government or commercial organization in medium robustness environments. Field devices monitor and control instruments in DCS and SCADA systems used in oil and gas pipelines, electric generation and transmission, chemical manufacturing, water treatment and many other critical infrastructure processes.

The Protection Profile defines:

- Assumptions about the security aspects of the environment in which a SCADA field device will be used;
- Threats that are to be addressed by the TOE;

- Security objectives for the TOE and its environment;
- Functional and assurance requirements to meet those security objectives; and
- Rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

1.3 Conventions

The notation, formatting, and conventions used in this Protection Profile are largely consistent with Version 2.3 of the Common Criteria. Presentation choices discussed in this section are for the aid of the reader. The Common Criteria allows several operations to be performed on functional requirements; refinement, selection and assignment are used in this Protection Profile.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. **Refinement** of security requirements is denoted by the work “Refinement” in **bold text** after the element number and the additional text in the requirement is displayed as **bold text**.

Refinement example:

Original:

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Refinement:

FMT_SMR.1.2 **Refinement:** The TSF shall be able to associate users with **defined security** roles.

The *selection* operation is used to select one or more options provided by the Common Criteria in stating a requirement. Selections that have been made by the Protection Profile authors are denoted by *italicized text* in brackets, selections to be filled in by the Security Target author appear in square brackets with an indication that a selection is to be made, [selection:].

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 Protection Profile authors are denoted by showing the value in square brackets, [Assignment_value], assignments that are to be filled by the Security Target author appear in square brackets with an indication that an assignment is to be made [assignment:].

Selection and Assignment example:

Original:

FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear*, [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorized identified roles*].

Selection and Assignments made:

FMT_MTD.1.1 The TSF shall restrict the ability to [*change_default, modify, delete, [view]*] the [security related data] to [authorized users].

This Protection Profile also uses National Information Assurance Partnership (NIAP) interpretations 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).

Naming Conventions

Assumptions: TOE security environment assumptions are given names beginning with “A.” followed by a descriptive label all in caps - - e.g., A.ADMINISTRATION.

Threats: TOE security environment threats are given names beginning with “T.” followed by a descriptive label all in caps - - e.g., T.SIGNAL_DETECT.

Policy Statements: Policy statements are given names beginning with “P.” followed by a descriptive label all in caps - - e.g., P.PHYSICAL_ACCESS.

Security Objectives for the TOE: Security Objectives are given names beginning with “O.” followed by a descriptive label all in caps - - e.g., O.ACCESS.

Security Objectives for the Environment: Security Objectives for the Environment are given names beginning with “OE.” followed by a descriptive label all in caps - - e.g., OE.ACCESS.

1.4 Document Organization

Section 1 introduces this Protection Profile document.

Section 2 describes the TOE and the environment.

Section 3 specifies TOE assumptions, threats and organizational security policies.

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

Section 5 specifies the functional and assurance requirements for the TOE.

Section 6 provides the rationale for the security objectives and the security requirements. The objectives rationale shows the security objectives address the threats and policies. The requirements rationale shows that the requirements meet the objectives and that all dependencies are satisfied. In addition, rationale is provided for the Strength of Function (SOF) and Assurance requirements.

2 Target of Evaluation (TOE) Description

This Protection Profile specifies the minimum security requirements for a Target of Evaluation (TOE) that is a SCADA field device. Common functions of a SCADA field device include:

- Collecting measurements from sensors
- Making logic and control calculations
- Issuing control commands that modify a process
- Communicating with an IT application

Examples of product categories that would be included in this TOE description are programmable logic controllers (PLC's), remote terminal units (RTU's), programmable automation controllers (PAC's), and intelligent electronic devices (IED's). These field devices are typically found in remote sites in SCADA networks such as pumping plants, substations, or turnouts.

The functionality of a field device can vary a great deal. Sophisticated field devices can run programs and control complex processes. Simple field devices can be limited to a small number of measurements and controls. This Protection Profile is applicable to any field device without regard to the amount of measurement, calculation or control that takes place in the device.

Field devices can communicate with directly connected HMI or SCADA application servers in a control center. The Protection Profile is applicable to any field device without regard to the location or type of IT application that communicates with the field device.

While the title of this Protection Profile refers to SCADA field devices, it may be applicable to similar field devices used in a DCS or any other control or monitoring system. In fact many field devices that are used in SCADA systems are also used in DCS and PLC based control systems.

The TOE includes all resident software, hardware, and firmware in a field device. The communication path and channels to the TOE are not part of the TOE. A simple way to describe the TOE boundary is the physical boundary around the hardware platform. For example, the TOE boundary could be the field device case for a monolithic Field Device or the rack or base for a component field device. The TOE boundary begins when data arrives at a physical interface and ends when data leaves a physical interface.

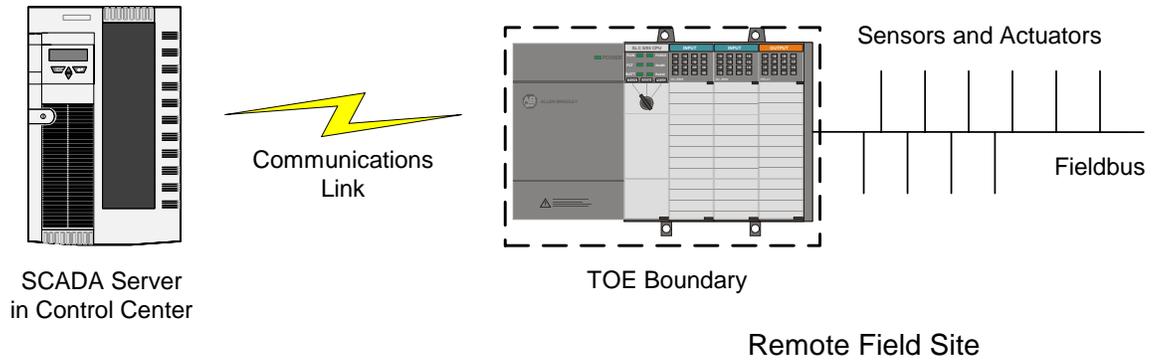


Figure 1 – TOE Boundary

Users are outside of the TOE boundary. They do however interact with the TOE through TOE Security Functions (TSF). Examples of this are user authentication for TOE management and access control requirements based on a user id or role.

Similarly, external IT entities are outside of the TOE boundary. A HMI or SCADA control server may communicate with the TOE, but these external IT entities and their communication links to the TOE are not in the TOE. The Protection Profile does have requirements to validate the integrity and reasonableness of external information when it arrives inside the TOE.

A TOE responding to this Protection Profile can either be a monolithic TOE or a component TOE that is part of a larger SCADA system or subsystem composite TOE.

3 TOE Security Environment

3.1 Medium Robustness

The selection of a robustness level is based on the value of the resources, the authorization of entities, and the likelihood of an attempted attack.

- Value of the resources – a field device can control a portion of a critical infrastructure process and provides data that allows an Operator to maintain the proper function of a critical infrastructure process. It is considered between a medium and high value resource for the determining a robustness level.
- Authorization of entities – this criterion refers to the trustworthiness and access control of entities allowed to access, and potentially attack, a field device. Access to field devices is typically restricted to a small number of users consisting of Operators, Engineers, and System Administrators. Many organizations perform background checks on all users prior to granting access. This would place the field device in the “Fully Authorized” category as defined in the Common Criteria.

SCADA systems often communicate with field devices via a wide area network (WAN), and it is possible for an attacker to gain unauthorized access to the WAN. The ability to gain unauthorized access is based on the WAN technology used and the implementation. However, the possibility of an attacker gaining WAN access to attack a field device reduces the criterion to “Partially Authorized”.

- Likelihood of an attempted attack – there are factors that reduce the likelihood of attack on field devices. Field devices are typically on restricted networks that are not accessible from the Internet or even an organizations enterprise network; field device protocols, such as Modbus, DNP3, and Ethernet/IP, are attacked much less often than popular protocols found in enterprise networks, such as http, smtp, and sql; and tools and documentation required to attack SCADA systems is not readily available.

An attacker is likely to require SCADA skills and tools to attack the field devices. The frequency of attack on a field device is likely to be low, but attackers who target field devices may be highly skilled, highly motivated and have substantial financial resources.

The three factors described above equate to a Common Criteria selection for medium robustness. Additional information on the robustness decision along with graphs that map the criteria to robustness levels is available in Appendix D.

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 TOE's of medium robustness. Note that while highly sophisticated threat agents will not be

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

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).

3.2 Assumptions

The specific conditions below are assumed to exist in a TOE environment.

A.CORRECT_USER_ACTIONS	Authorized users and administrators are properly trained and will not take actions that intentionally affect the security of the TOE.
A.PHYSICAL_ACCESS	The TOE will be placed in a secure physical location which will prevent unauthorized physical access and modification.
A.PHYSICAL_ENVIRONMENT	The TOE will be placed in a physical environment that meets the manufacturer’s specifications for temperature, humidity, and other environmental factors. The TOE will be provided with power that meets the manufacturer’s specifications.
A.PROTECTED_CREDENTIALS	Authorized users and administrators will protect their login credentials from unauthorized disclosure.

3.3 Threats

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 Protection Profile. Threat agents are typically characterized by a number of factors such as expertise, available resources and motivation. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and

available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

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

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

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

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example) then expertise should be considered to be at the same “level” (low, medium, high, for example) as the resources because can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources). It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be “medium”. This normally indicates 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 supposed the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

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

The important general points we can make are:

- The motivation for threat agent defines the upper bound with respect to the level of robustness required for the TOE.
- A threat agent’s expertise and/or resources that is “lower” than the threat agent’s motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).
- The availability of attacks associate 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.

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

Table 1 – Medium Robustness Applicable Threats

Threat Name	Threat Definition
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's actions.
T.CRYPTO_COMPROMISE	A malicious user or process may cause key, data or executable code associate with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and data protected by those mechanisms.
T.FLAWED_DESIGN	Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.
T.FLAWED_IMPLEMENTATION	Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.

Version 0.71 – May 18, 2006

Threat Name	Threat Definition
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.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 transmitted during the course of legitimate use).
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (e.g., flooding the TOE with poll requests) via a resource exhaustion denial of service attack.
T.SPOOFING	A malicious user, process, or external 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.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

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.

3.4 Organizational Security Policies

The following table lists the Organizational Security Policies this Protection Profile.

Table 2 – Medium Robustness Applicable Organizational Security 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 to 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 communication channels.
P.CRYPTOGRAPHY	The TOE shall use NIST FIPS validated cryptography as a baseline with additional NSA approved methods for key management (i.e., generation, access, distribution, destruction, handling, and storage of keys) and cryptographic operations (i.e., encryption, decryption, signature, hashing, key exchange and random number generation services).
P.VULNERABILITY_ANALYSIS_TEST	The TOE must undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.

4 Security Objectives

4.1 TOE Security Objectives

This section defines the security objectives that are to be addressed by the TOE.

Table 3 – TOE Objectives

Objective Name	Objective Definition
O.ADMIN_ROLE	The TOE will provide administrator role 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 associate with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.
O.CHANGE_MANAGEMENT	The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.CRYPTO_RESIDUAL_INFORMATION	The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.

Version 0.71 – May 18, 2006

Objective Name	Objective Definition
O.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.DOCUMENT_KEY_LEAKAGE	The bandwidth of channels that can be used to compromise key material shall be documented.
O.MAINT_MODE	The TOE shall provide a mode from which recovery or initial startup procedures can be performed.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.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.RESOURCE_SHARING	The TOE shall provide mechanisms that mitigate attempts to exhaust the memory, computing and input/output resources provided by the TOE.
O.ROBUST_ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure delivery and management
O.ROBUST_TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.

Version 0.71 – May 18, 2006

Objective Name	Objective Definition
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.
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 Security Objectives for the Operating Environment

This section defines the security objectives that are to be addressed by the operating environment.

Table 4 – Objectives for the IT Environment

IT Environment Objective Name	Environment Objective Definition
OE.CORRECT_USER_ACTIONS	Authorized users and administrators will be trained to perform correct actions, and all users will undergo a periodic background check to determine suitability to be a user.
OE.PHYSICAL_ACCESS	Physical access to the TOE will be limited to authorized users.
OE.PHYSICAL_ENVIRONMENT	An appropriate environment, including power, temperature, humidity and other controls, will be maintained for the TOE.
OE.PROTECTED_CREDENTIALS	Authorized users and administrators will protect all login credentials for the TOE from exposure to other users and non-users.

5 TOE Security Requirements

This section provides functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE. These requirements consist of functional components from Part 2 of the Common Criteria and an Evaluation Assurance Level (EAL) containing assurance components from Part 3 of the Common Criteria.

5.1 TOE Security Functional Requirements

The functional security requirements for this Protection Profile, summarized in the following table 5-1, consist of components from Part 2 of the Common Criteria.

The statement of the TOE security requirements must include a minimum strength of function level for the TOE security functions. The minimum strength of function level for this Protection Profile is SOF-medium.

5.1.1 Security Functional Components

Table 5- Security Functional Requirements

Component	Component Name
FAU_ARP.1	Security Alarms
FAU_GEN.1-NIAP-0407	Audit Data Generation
FAU_GEN.2-NIAP-0410	User Identity Association
FAU_SAA.1-NIAP-0407	Potential Violation Analysis
FAU_SAR.1	Audit Review
FAU_SAR.2	Restricted Audit Review
FAU_SEL.1-NIAP-0407	Selective Audit
FAU_STG.2-NIAP-0429	Guarantees Of Audit Data Availability
FAU_STG.3	Action In Case Of Possible Audit Data Loss
FAU_STG.4	Prevention Of Audit Data Loss
FCS_BCM_(EXP).1	Baseline Cryptographic Module
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Distribution
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1	Cryptographic Operation
FDP_ACC.2	Complete Access Control

Version 0.71 – May 18, 2006

Component	Component Name
FDP_ACF.1-NIAP-0407	Security Attribute Based Access Control
FDP_DAU.1	Basic Data Authentication
FDP_ETC.2	Export Of User Data With Security Attributes
FDP_IFC.2	Complete Information Flow Control
FDP_IFF.1	Simple Security Attributes
FDP_ITC.2	Import Of User Data With Security Attributes
FDP_ROL.1	Basic Rollback
FDP_SDI.2	Stored Data Integrity Monitoring And Action
FIA_AFL.1	Authentication Failure Handling
FIA_ATD.1	User Attribute Definition
FIA_SOS.1	Verification Of Secrets
FIA_UAU.2	User Authentication Before Any Action
FIA_UAU.4	Single-use Authentication Mechanisms
FIA_UAU.7	Protected Authentication Feedback
FIA_UID.2	User Identification Before Any Action
FIA_USB.1	User-subject Binding
FMT_MOF.1	Management Of Security Functions Behavior
FMT_MSA.1	Management of Security Attributes
FMT_MSA.3	Static Attribute Initialization
FMT_MTD.1	Management Of TSF Data
FMT_MTD.2	Management Of Limits On TSF Data
FMT_MTD.3	Secure TSF Data
FMT_REV.1	Revocation
FMT_SMF.1	Specification Of Management Functions
FMT_SMR.2	Restriction on Security Roles
FPT_AMT.1	Abstract Machine Testing
FPT_FLS.1	Failure With Preservation Of Secure State
FPT_ITI.1	Inter-TSF Detection Of Modification
FPT_PHP.2	Notification Of Physical Attack
FPT_RCV.3	Automated Recovery Without Undue Loss

Component	Component Name
FPT_RPL.1	Replay Detection
FPT_RVM.1	Non-bypassability Of The TSP
FPT_SEP.2	Domain Separation
FPT_STM.1	Time Stamps
FPT_TST_(EXP).4	TSF Self-Test
FPT_TST_(EXP).5	Cryptographic Self-Test
FRU_PRS.2	Full Priority Of Service
FRU_RSA.1	Maximum Quotas
FTA_TAB.1	Default TOE Access Banners
FTA_TAH.1	TOE Access History
FTA_TSE.1	TOE Session Establishment
FTP_TRP.1	Trusted Path

5.1.2 Security Audit (FAU) Requirements

5.1.2.1 Security Audit Automatic Response (FAU_ARP.1)

FAU_ARP.1.1

The TSF shall take [an action of generating a real time security alarm that can be displayed on an HMI and is placed as an event in an audit log] upon detection of a potential security violation.

Application Note: The integrator of the TOE into a SCADA or DCS system will determine if the potential violation is displayed on one of the HMI displays. This decision may be made on by event type. This requirement merely makes it possible to display the potential violation.

The TOE does not require any automated action that would affect the availability of the TOE, such as blocking a user or resetting a TCP session because a false positive could affect availability for authorized uses and users.

5.1.2.2 Audit Data Generation (FAU_GEN.1-NIAP-0407)

FAU_GEN.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; and
- (b) All auditable events listed in Table 6;

- (c) [assignment: *events commensurate with a basic level of audit introduced by the inclusion of explicit requirements determined by the Security Target author*].

FAU_GEN.1.2-NIAP-0407

Refinement: 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), **the user 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 Protection Profile / Security Target, [information specified in column three of Table 6 below].
- (c) **An indicator that the audit event is a security event.**

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

The refinement requires the ST to designate events as security events to help monitoring products and services extract security events from the audit logs. A Security Target may have further define the security events into categories such as denial of service, authentication failure, or protocol violation.

Table 6 – Auditable Events

Component	Auditable Events	Additional Audit Records Contents
FAU_ARP.1	None	None
FAU_GEN.1-NIAP-0407	None	None
FAU_GEN.2-NIAP-410	None	None
FAU_SAA.1-NIAP-0407	Enabling, disabling, and modifying any of the analysis mechanisms	The configurable threshold value
FAU_SAR.1	Reading of information from the audit records.	None

Version 0.71 – May 18, 2006

Component	Auditable Events	Additional Audit Records Contents
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	None
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating.	List of what was included or excluded
FAU_STG.1-NIAP-0429	None	None
FAU_STG.3	Actions taken due to exceeding of a threshold	None
FAU_STG.4	Actions taken due to the audit storage failure	None
FCS_BCM_(EXP).1	None	None
FCS_CKM.1 FCS_CKM.2 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_COP.1	Success and failure, and the type of cryptographic operation.	Any applicable cryptographic mode(s) of operation, subject attributes and object attributes
FDP_ACC.2	None	None
FDP_ACF.1-NIAP-0407	All requests to perform an operation on an object covered by the SFP	None
FDP_DAU.1	Successful and unsuccessful generation of validity evidence	None
FDP_ETC.2	All attempts to export information	None
FDP_IFF.1	All decisions on requests for information flow	None
FDP_IFC.2	None	None
FDP_ITC.2	All attempts to import user data, including any security attributes	None

Version 0.71 – May 18, 2006

Component	Auditable Events	Additional Audit Records Contents
FDP_ROL.1	All attempts to perform rollback operations	None
FDP_SDI.2	All attempts to check the integrity of user data, including an indication of the results of the check, if performed	None
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken and the subsequent, if appropriate, restoration to the normal state	None
FIA_ATD.1	None	None
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	None
FIA_UAU.2	All use of the authentication mechanism	Indication of success or failure
FIA_UAU.4	Attempts to reuse authentication data	None
FIA_UAU.7	None	None
FIA_UID.2	All use of the user identification mechanism, including the user identity provided	None
FIA_USB.1	Success and failure of binding of user security attributes to a subject (e.g. success and failure to create a subject)	Indication of success or failure
FMT_MOF.1	All modifications in the behavior of the functions in the TSF	None
FMT_MSA.1	All modifications of the values of security attributes	None
FMT_MSA.3	Modifications of the default setting of restrictive rules All modifications of the initial values of security attributes	None

Version 0.71 – May 18, 2006

Component	Auditable Events	Additional Audit Records Contents
FMT_MTD.1	All modifications to the values of the of the TSF data	None
FMT_MTD.2	All modifications to the limits on TSF data All modifications in the actions to be taken in case of violation of the limits	None
FMT_MTD.3	All rejected values of TSF data	None
FMT_REV.1	All attempts to revoke security attributes	None
FMT_SMF.1	Use of the management functions	None
FMT_SMR.2	Modifications to the group of users that are part of a role Unsuccessful attempts to use a role due to the given conditions on the roles	None
FPT_AMT.1	Execution of the tests of the underlying machine and the results of the tests	None
FPT_FLS.1	Failure of the TSF	None
FPT_ITL.1	The detection of modification of transmitted TSF data	None
FPT_PHP.2	Detection of intrusion	None
FPT_RCV.3	The fact that a failure or service discontinuity occurred and the type of failure or service discontinuity The resumption of the regular operation	None
FPT_RPL.1	Detected replay attacks	None
FPT_RVM.1	None	None
FPT_SEP.2	None	None
FPT_STM.1	Changes to the time	None

Component	Auditable Events	Additional Audit Records Contents
FPT_TST_(EXP).4	Execution of the self tests and the results of the tests	None
FPT_TST_(EXP).5	Execution of the self tests and the results of the tests	None
FRU_PRS.2	All attempted uses of the allocation function which involves the priority of the service functions	None
FRU_RSA.1	All attempted uses of the resource allocation functions for resources that are under control of the TSF	None
FTA_TAB.1	None	None
FTA_TSE.1	All attempts at establishment of a user session	None
FTA_TRP.1	All attempted uses of the trusted path functions Identification of the user associated with all trusted path invocations, if available	None

5.1.2.3 User Identity Association (FAU_GEN.2-NIAP-0410)

FAU_GEN.2.1-NIAP-0410

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

Application Note: This requirement recognizes the difference between actions initiated by a user and an IT system, such as a control server. When a user, typically an Operator or Administrator, initiates the action their userID must be logged.

5.1.2.4 Potential Violation Analysis (FAU_SAA.1-NIAP-0407)

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

- (a) An Administrator specified number of authentication failures;
- (b) Any detected replay of TSF data or security attributes;
- (c) Any failure of the cryptographic self-tests;
- (d) Any failure of the other TSF self-tests;
- (e) An Administrator specified number of encryption failures;
- (f) An Administrator specified number of decryption failures;
- (g) [An Administrator specified number of cryptographic data integrity verification failures]; and
- (h) [assignment: *additional events from the set of defined auditable events*].

Application Note: The intent of this requirement is that an alarm is generated (FAU_ARP.1) once the threshold for an event is met. Once the alarm has been generated it is assumed that the “count” for that event is reset to zero. The Administrator configurable number of authentication failures in (a) is intended to be the same value as specified in FIA_AFL.1.

The failure of TSF self-tests in (d) include failures of FPT_TST _(EXP).

5.1.2.5 Audit Review (FAU_SAR.1)

FAU_SAR.1.1

The TSF shall provide [authorized Administrators] with the capability to read [security related audit information] from the audit records.

FAU_SAR.1.2

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

Application Note: Only users in the Administrator role are allowed to read security related audit records. FAU_GEN.1.2-NIAP-0407 required the TOE to have an indicator in each record for audit events. This could be implemented as a log event category field.

There are no restrictions in this Protection Profile on authenticated users reading any of the other TOE audit logs. This may be required by Operators or Maintenance personnel to deal with normal process incidents at a field site.

5.1.2.6 Restricted Audit Review (FAU_SAR.2)

FAU_SAR.2.1

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

5.1.2.7 Selective Audit (FAU_SEL.1-NIAP-0407)

FAU_SEL.1-NIAP-0407

Refinement: The TSF shall **allow only the Administrators** to include or exclude auditable events from the set of audited events based on the following attributes:

- (a) user identity;
- (b) event type;
- (c) [*object identity, subject identity, host identity*];
- (d) success of auditable security events;
- (e) failure of auditable security events; and
- (f) [assignment: *list of additional criteria that audit selectivity is based upon*].

Application Note: “event type” is to be defined by the Security Target author. The TOE requires a security event type category, but further categorization is possible in the Security Target. For example security event sub-categories could be identified such as denial of service, access control, replay and spoofing, etc. The intent is to be able to include or exclude classes of audit events.

5.1.2.8 Guarantees Of Audit Data Availability (FAU_STG.2-NIAP-0429)

FAU_STG.2.1-NIAP-0429

Refinement: The TSF shall **restrict the deletion of stored** audit records in the audit trail **to Administrators**.

FAU_STG.2.2-NIAP-0429

The TSF shall be able to [*prevent*] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.2.3-NIAP-0429

The TSF shall ensure that [one hour] audit records will be maintained when the following conditions occur: [*audit storage exhaustion, failure, attack*].

Application Note: Administrators are the only users allowed to delete records in the TOE, but even Administrators are not allowed to modify audit records.

FAU_STG.2.3 – requires the TOE to have some secondary method of storing a short period, one hour, of audit logs in case of an audit log failure.

5.1.2.9 Prevention Of Audit Data Loss (FAU_STG.3)

FAU_STG.3.1

The TSF shall take [an action to log an alarm in the audit log and [assignment: *other actions to be taken in case of possible audit storage failure*]] if the audit trail exceeds [an Administrator settable percentage of storage capacity].

5.1.2.10 Prevention Of Audit Data Loss (FAU_STG.4)

FAU_STG.4.1

The TSF shall [*overwrite the oldest stored audit records*] and [write an alarm to the audit log] if the audit trail is full.

Application Note: The SCADA market prefers the latest data be retained in case the audit log is full and will sacrifice older, historical data if necessary. This reflects the priority of availability over a forensic capability.

The alarm generated in FAU_STG.3.1 should provide an Administrator with time to address the potential loss of audit log data due to a near term full log condition.

5.1.3 Cryptographic Support (FCS) Requirements

5.1.3.1 Baseline Cryptographic Module (FCS_BCM_(EXP).1)

FCS_BCM_(EXP).1.1

All cryptographic modules shall comply with FIPS PUB 140-2 when performing FIPS-approved cryptographic functions in FIPS-approved cryptographic modes of operation.

Application Note: This Protection Profile was sponsored by the US Government NIST (through the PCSRf) and is likely to be submitted to the NIAP CCEVS program. These two facts mandate the above specification for cryptographic modules. Organizations may choose to modify this Protection Profile or a Security Target to specify an industry standard or different national algorithm.

FCS_BCM_(EXP).1.2

Cryptographic functions and cryptographic modes of operation as identified in this Protection Profile shall be NSA-validated.

Application Note: In time, Operating System Protection Profile requirements are expected to evolve such that NSA-validated cryptographic modules shall only contain cryptographic functions, cryptographic modes of operation, and other types of cryptographic processing that are compliant with this Protection Profile.

FCS_BCM_(EXP).1.3

All cryptographic modules implemented in the TCSF [selection:

- Entirely in hardware shall have a minimum overall rating of FIPS PUB 140-2, Level 3;
- Entirely in software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS 140-2, Level 3 for the following: Cryptographic Module Ports and Interfaces; Roles, Services and Authentication; Cryptographic Key Management; Design Assurance; and FIPS PUB 140-2, Level 4 Self Tests as defined by this Protection Profile;
- 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; Design Assurance; and FIPS PUB 140-2, Level 4 Self Tests as defined in this Protection Profile.]

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 a “combination of hardware and software”.

5.1.3.2 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *FIPS 140-2 approved key generation algorithm*] and specified cryptographic key sizes [assignment: *FIPS 140-2 approved key size for symmetric/private key cryptographic algorithms and asymmetric/public key algorithms*] that meet the following standards: [FIPS-140-2].

5.1.3.3 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 [assignment: *FIPS 140-2 approved key distribution method*] that meet the following standards: [FIPS 140-2].

Application Note: This Protection Profile was sponsored by the US Government NIST (through the PCSRF) and is likely to be submitted to the NIAP CCEVS program. These two facts mandate the above specification for cryptographic modules. Organizations may choose to modify this Protection Profile or a Security Target to specify an industry standard or different national algorithm.

5.1.3.4 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified key destruction method [zeroization] that meets the following: [FIPS 140-2 Level 2 for Subscriber/Level 3 for Registration and Certification Authorities].

Application Note: Zeroization destroys unencrypted private keys by altering and deleting memory and storage containing the keys.

5.1.3.5 Cryptographic Operation (FCS_COP.1)

FCS_COP.1.1

The TSF shall perform [digital signature generation and verification, message authentication, encryption and decryption, and key exchange or negotiation] with a specified cryptographic algorithm [assignment: *FIPS 140-2 cryptographic algorithms*] and cryptographic key sizes [assignment: *FIPS 140-2 approved key size for symmetric/private key cryptographic algorithms and asymmetric/public key algorithms*] that meet the following standards: [FIPS 140-2].

5.1.4 User Data Protection (FDP) Requirements

5.1.4.1 Complete Access Control (FDP_ACC.2)

FDP_ACC.2.1

The TSF shall enforce the [P.Access_Control SFP] on: [all subjects representing a user in the Administrator, Operator, and Display roles, all objects, and [assignment: *list of subjects and objects covered by the SFP*]] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2

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

5.1.4.2 Security Attribute Based Access Control (FDP_ACF.1-NIAP-0407)

FDP_ACF.1.1-NIAP-0407

The TSF shall enforce the [P.Access_Control SFP] to objects based on [role, location, time of day / day of week, and [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]].

FDP_ACF.1.2-NIAP-0407

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

FDP_ACF.1.3-NIAP-0407

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

FDP_ACF.1.4-NIAP-0407

The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

5.1.4.3 Basic Data Authentication (FDP_DAU.1)

FDP_DAU.1.1

The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [

- (a) audit logs
- (b) process data
- (c) configuration data
- (d) user data
- (e) [assignment: *list of objects or information types*].

FDP_DAU.1.2

The TSF shall provide the [subjects corresponding to users in an Administrator role and [assignment: *list of subjects*]] with the ability to verify evidence of the validity of the indicated information.

Application Note: FDP_DAU.1 requires the TOE to provide evidence of integrity. Audit logs and configuration data are easily understood. Process data could be writes to the TOE from a HMI or data coming from an instrument. User data could be userID's and authentication credential storage.

5.1.4.4 Export Of User Data With Security Attributes (FDP_ETC.2)

FDP_ETC.2.1

The TSF shall enforce the [P.Access_Control SFP] when exporting user data, controlled under the SFP, outside of the TSC.

FDP_ETC.2.2

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3

The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.

FDP_ETC.2.4

The TSF shall enforce the following rules when user data is exported from the TSC: [security attributes shall be included that will allow a recipient to verify the data came from the user and has not been modified in transit and [assignment: *additional exportation control rules*]].

Application Note: Export user data may be required to send data to a management system or to use as a template for configuring other field devices.

5.1.4.5 Complete Information Flow Control (FDP_IFC.2)

FDP_IFC.2.1

The TSF shall enforce the [P.Access_Control SFP] on [all subjects and information within a TOE] and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2

The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

5.1.4.6 Simple Security Attributes (FDP_IFF.1)

FDP_IFF.1.1

The TSF shall enforce the [P.Access_Control SFP] based on the following types of subject and information security attributes:

- [
- source subject identifier
 - user role related to source subject identifier
 - [assignment: *list of additional subjects and information controlled under the indicated SFP and related security attributes*]]

FDP_IFF.1.2

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].

FDP_IFF.1.3

The TSF shall enforce the following: [assignment: *additional information flow control SFP rules*].

FDP_IFF.1.4

The TSF shall provide the following: [assignment: *list of additional SFP capabilities*].

FDP_IFF.1.5

The TSF shall explicitly authorize an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorize information flows*].

FDP_IFF.1.6

The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

5.1.4.7 Import Of User Data With Security Attributes (FDP_ITC.2)

FDP_ITC.2.1

The TSF shall enforce the [P.Access_Control SFP] when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.2.2

The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4

The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: *additional importation control rules*].

5.1.4.8 Basic Rollback (FDP_ROL.1)

FDP_ROL.1.1

The TSF shall enforce [P.Access_Control SFP] to permit the rollback of the [TOE security configuration, TOE security management, and [assignment: *list of operations*]].

FDP_ROL.1.2

The TSF shall permit operations to be rolled back within [the last three management or configuration changes].

Application Note: An Administrator may unintentionally make a change that affects the security of the TOE. Rollback provides a fast means for recovery.

5.1.4.9 Stored Data Integrity Monitoring And Action (FDP_SDI.2)

FDP_SDI.2.1

The TSF shall monitor user, subject and object data stored within the TSC for [corruption of data] on all objects based on the following attributes: [assignment: user data attributes].

FDP_SDI.2.2

Upon detection of a data integrity error, the TSF shall [write a log entry to a log and [assignment: *action to be taken*]].

Application Note: User, subject and object data in a field device includes the points database structure and values as well as the configuration of the field device for a particular field site. This requirement will identify when this data used for the proper operation of the field device is corrupted, unintentionally or maliciously.

5.1.5 Identification And Authentication (FIA) Requirements

5.1.5.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1

The TSF shall detect when [an administrator configurable positive integer within [a Security Administrator configurable amount of time]] unsuccessful authentication attempts occur related to [a user's authentication].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [generate a security event in the audit log].

Application Note: Availability is the most critical security objective in SCADA systems. Locking an account due to exceeding failed authentication attempt thresholds is not typically recommended.

5.1.5.2 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [

- (a) unique userID
- (b) data required to verify authentication credentials
- (c) user roles
- (d) time and date when user account is to be disabled
- (e) time and day of the week the user is allowed to login to the TSF
- (f) [assignment: *list of additional security attributes*]].

Application Note: The TSF must support granting unique userID's so users are not forced to share userID's and credentials. The remaining user attributes play a part in access control decisions.

The "time and date when user account is to be disabled" is useful for contractors or employees that need short term access. This requirement also helps to remove access for a pre-planned job change such as retirement.

5.1.5.3 TSF Verification Of Secrets (FIA_SOS.1)

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets meet: [

- (a) a two-factor authentication requirement for users in an Administrator role
- (b) a password complexity standard that is configurable by an Administrator for users in an Operator or Display role. The default complexity standard must be at least eight characters long and contains at least one letter, one number and one non-alphanumeric character.]

Application note: Two-factor authentication includes two of the following three authentication factors: something you know (such as a password or PIN), something you have (such as a token or smart card), and something you are (such as a fingerprint or hand geometry).

This requirement also enforces password complexity requirements for users in the Operator and Display roles. There is a default complexity standard, but this must be configurable by the Administrator.

5.1.5.4 User Authentication Before Any Action (FIA_UAU.2)

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.

Application Note: This requirement, in conjunction with FIA_UID.2, prevents any action on the TSF prior to identification and authentication.

5.1.5.5 Single-use Authentication Mechanisms (FIA_UAU.4)

FIA_UAU.4.1

The TSF shall prevent reuse of authentication data related to [at least one of the factors in the two-factor authentication required for users in an Administrator role].

Application Note: The TSF must prevent the reuse of authentication credentials. One way this can be accomplished through one of the many challenge / response protocols and ensuring a large and random challenge size.

5.1.5.6 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1

The TSF shall provide only [an indication the authentication is in progress, succeeded or failed] to the user while the authentication is in progress.

Application Note: The TSF must not indicate if the userID or credential was correct or incorrect. Similarly, the TSF must not indicate if a smart card, fingerprint, or PIN is correct or incorrect. Only the final result of the authentication attempt is presented to the user.

This prevents an attacker from learning if he or she guessed one component of authentication correctly.

5.1.5.7 User Identification Before Any Action (FIA_UID.2)

FIA_UID.2.1

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

5.1.5.8 User-subject Binding (FIA_USB.1)

FIA_USB.1.1

The TSF shall associate the following user security attributes with subjects acting on behalf of that user: [userID, user role(s) and [assignment: *list of additional user security attributes*].

FIA_USB.1.2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *additional rules for the initial association of attributes*].

FIA_USB.1.3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [changes will not take affect until the user logs out and logs in again and [assignment: *additional rules for the changing of attributes*].

Application Note: The userID and corresponding user roles determines if access and actions on subjects are permitted. The other criteria in FIA_ATD.1 are used only to determine if login is permitted.

5.1.6 Security Management (FMT) Requirements

5.1.6.1 Management Of Security Functions Behavior (FMT_MOF.1)

FMT_MOF.1.1

The TSF shall restrict the ability to [*determine the behavior of, disable, enable, or modify the behavior of*] the functions: [

- (a) Authentication functions
- (b) Authorization functions
- (c) Auditing functions
- (d) Data integrity functions
- (e) Non-repudiation functions
- (f) Controlled connection-oriented resource allocation (FRU_RSA.1) parameters
- (g) [assignment: additional *list of functions*]

to [the Administrator role, [assignment: *additional authorized roles*].

Application Note: The Security Target may define a more granular set of Administrator roles. For example, an Auditor role may be defined that would have the ability to modify the behavior of audit functions.

5.1.6.2 Management Of Security Attributes (FMT_MSA.1)

FMT_MSA.1.1

The TSF shall enforce the [P.Access_Control SFP] to restrict the ability to [*change_default, modify, or delete*] the security attributes [that are restricted to the Administrator role in Table 7, [assignment: *additional security attributes*]] to [the Administrator role].

5.1.6.3 Static Attribute Initialization (FMT_MSA.3)

FMT_MSA.3.1

The TSF shall enforce the [P.Access_Control SFP] to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

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

Application Note: The default values for security attributes must be set to a restrictive default value so the TOE is secure from the start. Administrators are allowed to accept the risk of a less secure setting and change these default values.

5.1.6.4 Management Of TSF Data (FMT_MTD.1)

FMT_MTD.1.1

The TSF shall restrict the ability to [*change_default, modify, delete, and clear*] the [audit records specified in FAU_GEN.1, TSF configuration, database configuration and [assignment: *list of TSF data*]] to [the Administrator role].

5.1.6.5 Management Of Limits On TSF Data (FMT_MTD.2)

FMT_MTD.2.1

The TSF shall restrict the specification of the limits for [audit trails specified in FAU_GEN.1, quotas on controlled connection-oriented resources and [assignment: *list of TSF data*]] to [the Administrator role].

FMT_MTD.2.2

The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [overwrite the oldest stored audit record and enter an event in the audit log, [assignment: *actions to be taken*]].

Application Note: For FMT_MTD.2.2, the Security Target author should specify the actions that the TOE takes for each controlled connection-oriented resource when the quota established by the Administrator is reached.

5.1.6.6 Secure TSF Data (FMT_MTD.3)

FMT_MTD.3.1

The TSF shall ensure that only secure values are accepted for TSF data.

5.1.6.7 Revocation (FMT_REV.1)

FMT_REV.1.1

The TSF shall restrict the ability to revoke security attributes associated with the [users, subjects, objects, [assignment: *other additional resources*]] within the TSC to [the Administrator role].

FMT_REV.1.2

The TSF shall enforce the rules [within an Administrator configurable time of the revocation and [assignment: *specification of revocation rules*]].

Application Note: Security attributes include the set of authorization rights given to a user or assigned to an object. These rights may need to be revoked and this revocation enforced on a timely basis. For example, a suspected attack from a specific userID may require the revocation of the authorization rights associated with that userID.

5.1.6.8 Specification of Management Functions

FMT_SMF.1

The TSF shall be capable of performing the following security management functions: [management functions in Table 7 and [assignment: *list of security management functions to be provided by the TSF*]].

Table 7 – Management Functions

Component	Management Functions
FAU_ARP.1	None
FAU_GEN.1-NIAP-0407	None
FAU_GEN.2-NIAP-0410	None
FAU_SAA.1-NIAP-0407	Maintenance of the rules by (adding, modifying, deletion) of the rules
FAU_SAR.1	None
FAU_SAR.2	None
FAU_SEL.1-NIAP-0407	None
FAU_STG.2	Maintenance of the parameters that control the audit storage capability
FAU_STG.3	Maintenance of the threshold Maintenance (deletion, modification, addition) of actions to be taken in case of imminent audit storage failure.
FAU_STG.4	Maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure
FCS_BCM_(EXP).1	None
FCS_CKM.1 FCS_CKM.2 FCS_CKM.4	The management of changes to cryptographic key attributes. Examples of key attributes include user, key type, validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption).
FCS_COP.1	None
FDP_ACC.2	None
FDP_ACF.1-NIAP-0407	Managing the attributes used to make explicit access or denial based decisions
FDP_DAU.1	The assignment or modification of the objects for which data authentication may apply could be configurable in the system
FDP_ETC.2	The additional exportation control rules could be configurable by a user in a defined role
FDP_IFC.2	None
FDP_IFF.1	Managing the attributes used to make explicit access based decisions

Component	Management Functions
FDP_ITC.2	The modification of the additional control rules used for import
FDP_ROL.1	The boundary limit to which rollback may be performed could be a configurable item within the TOE Permission to perform a rollback operation could be restricted to a well defined role
FDP_SDI.2	The actions to be taken upon detection of an integrity error could be configurable
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts Management of actions to be taken in the event of an authentication failure
FIA_ATD.1	Management of the user access control parameters stated in this requirement
FIA_SOS.1	The management of the metric used to verify the secrets
FIA_UAU.2	Management of the authentication data by an administrator Management of the authentication data by the user associated with this data
FIA_UAU.4	None
FIA_UAU.7	None
FIA_UID.2	The management of the user identities
FIA_USB.1	An authorized administrator can define default subject security attributes An authorized administrator can change subject security attributes
FMT_MOF.1	Managing the group of roles that can interact with the functions in the TSF
FMT_MSA.1	Managing the group of roles that can interact with the security attributes
FMT_MSA.3	Managing the group of roles that can specify initial values Managing the permissive or restrictive setting of default values for a given access control SFP

Version 0.71 – May 18, 2006

Component	Management Functions
FMT_MTD.1	Managing the group of roles that can interact with the TSF data
FMT_MTD.2	Managing the group of roles that can interact with the limits on the TSF data
FMT_MTD.3	None
FMT_REV.1	<p>Managing the group of roles that can invoke revocation of security attributes</p> <p>Managing the lists of users, subjects, objects and other resources for which revocation is possible</p> <p>Managing the revocation rules</p>
FMT_SMF.1	None
FMT_SMR.1	Managing the group of users that are part of a role
FPT_AMT.1	<p>Management of the conditions under which abstract machine test occurs, such as during initial start-up, regular interval, or under specified conditions</p> <p>Management of the time interval if appropriate</p>
FPT_FLS.1	None
FPT_ITI.1	None
FPT_PHP.2	<p>Management of the user or role that gets informed about intrusions</p> <p>Management of the list of devices that should inform the indicated user or role about the intrusion</p>
FPT_RCV.3	<p>Management of who can access the restore capability within the maintenance mode</p> <p>Management of the list of failures/service discontinuities that will be handled through the automatic procedures</p>
FPT_RPL.1	<p>Management of the list of identified entities for which replay shall be detected</p> <p>Management of the list of actions that need to be taken in case of replay</p>
FPT_RVM.1	None
FPT_SEP.2	None
FPT_STM.1	Management of the time

Component	Management Functions
FPT_TST_(EXP).4	Management of the time interval for self tests and who can perform on demand self-tests
FPT_TST_(EXP).5	Management of the time interval for self tests and who can perform on demand self-tests
FRU_PRS.2	Assignment of priorities to each subject in the TSF
FRU_RSA.1	Specifying maximum limits for a resource for groups and/or individual users and/or subjects by an administrator
FTA_TAB.1	Maintenance of the banner by the authorized administrator
FTA_TSE.1	Management of the session establishment conditions by the authorized administrator
FTA_TRP.1	Configuring the actions that require trusted path, if supported

5.1.6.9 Restriction On Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles [Administrator, Operator, Display, and [assignment: *any other roles*]].

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 [the Administrator role shall be able to administer the TOE locally; the Administrator role shall be able to administer the TOE remotely; a userID is associated with only one role; [assignment: *conditions for the different roles*]].

Application Note: The CCEVS Consistency Instruction Manual for Medium Robustness environments recommends the Administrator functions be separated into Security Administrator, Cryptographic Administrator and Audit Administrator roles. This is not practical for the SCADA environment that typically does not have the staff to enforce that type of separation of duties.

5.1.7 Protection Of The TSF (FPT) Requirements

5.1.7.1 Abstract Machine Testing (FPT_AMT.1)

FPT_AMT.1.1

Refinement: The TSF shall run a suite of tests [*during initial start-up, periodically during normal operation, at the request of an authorized administrator, [assignment: other conditions]*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the **software portions of the TSF**.

Application Note: The test suite need only cover aspects of the underlying abstract machine on which the TSF relies to implement the required functions, including domain separation.

5.1.7.2 Failure With Preservation Of Secure State (FPT_FLS.1)

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur: [reboot, absence or loss of available of computing memory or storage, [assignment: *list of types of failures in the TSF*]].

5.1.7.3 Inter-TSF Detection of Modification (FPT_ITI.1)

FPT_ITI.1.1

The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: [modification of one or more bits].

FPT_ITI.1.2

The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform [the actions of discarding modified information and entering a corrupt data event in the audit log] if modifications are detected.

Application Note: The integrity of data exported from the TSF is required; the confidentiality of this data is not required. This data could be used to configure similar field devices or for another management function. Loss of integrity of a single bit must be identified, and any loss of integrity will generate an event in the audit log.

5.1.7.4 Notification Of Physical Attack (FPT_PHP.2)

FPT_PHP.2.1

The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.2.2

The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FPT_PHP.2.3

For [the TSF enclosure around TSF data processing and storage components and [assignment: *list of TSF devices/elements for which active detection is required*]], the TSF shall monitor the devices and elements and notify [an Administrator] when physical tampering with the TSF's devices or TSF's elements has occurred.

Application Note: The FIPS 140-2 requirements for a crypto module will carry with in the tamper evident features of the FIPS 140-2 standard.

5.1.7.5 Automated Recovery Without Undue Loss (FPT_RCV.3)

FPT_RCV.3.1

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

FPT_RCV.3.2

For [all shutdowns, reboots, and [assignment: *list of failures/service discontinuities*]], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3

The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [an Administrator configurable time period with a default value of two minutes of operational information] for loss of data or objects within the TSC.

FPT_RCV.3.4

The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

5.1.7.6 Replay Detection (FPT_RPL.1)

FPT_RPL.1.1

The TSF shall detect replay for the following entities: [authentication data, TSF data, security attributes, [assignment: *list of identified entities*]].

FPT_RPL.1.2

The TSF shall perform [reject data; create audit event; and [assignment: *list of specific actions*]] when replay is detected.

5.1.7.7 Non-bypassability Of The TSP (FPT_RVM.1)

FPT_RVM.1.1

The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Application Note: The TSF will not perform a mediated action prior to the security functions being available. This prevents attacks during a reboot or other initialization sequence.

5.1.7.8 Domain Separation (FPT_SEP.2)

FPT_SEP.2.1

The non-isolated portion of the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.2.2

The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.2.3

Refinement: The TSF shall maintain separation of the part of the TSF related to **cryptography** that protects it from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to cryptography.

5.1.7.9 Time Stamps (FPT_STM.1)

FPT_STM.1.1

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

5.1.7.10 TSF Self Test (FPT_TST_(EXP).4)

FPT_TST_(EXP).4.1

The TSF shall run a suite of self tests [during initial start-up, periodically during normal operation as specified by an Administrator and at the request of an Administrator] to demonstrate the correct operation of the hardware portions of the TSF.

FPT_TST.1.2_(EXP).4.2

The TSF shall provide the Administrator role with the capability to use a TSF-provided cryptographic function to verify the integrity of [all TSF data except the following: audit data, [assignment: *other dynamic TSF data for which no integrity validation is justified*]].

FPT_TST.1.3_(EXP).4.3

The TSF shall provide the Administrator role with the capability to use a TSF-provided cryptographic function to verify the integrity of stored TSF executable code.

Application Note: The explicit requirement is necessary since some TOE data are dynamic (e.g., data in the audit trail, passwords) and so interpretation of

“integrity” for FPT_TST.1.2 is required, leading to potential inconsistencies. The intention is that any parameter that only an administrator can control is verified to ensure its integrity is maintained. It is not necessary for the TOE to verify the integrity of audit data or user’s passwords. If the TOE verifies the integrity of these, the Security Target author may fill in the assignment to include them.

Since this TOE includes all the hardware necessary for the operation of the TOE, the element FPT_TST_(EXP).4.1 ensures that the hardware aspects of the TOE are tested prior to or during operations. It is not necessary to test the software portions of the TSF, since the evaluation ensures the correct operation of the software, software does not degrade or suffer intermittent faults, as does hardware, and integrity of the software portions of the TSF are addressed by FPT_TST_(EXP).4.3. Note that since cryptographic functions implemented in hardware that are part of a cryptomodule are tested in FPT_TST_(EXP).5, this requirement only applies to cryptographic functionality implemented in hardware that is not implemented in a cryptomodule (for instance, an implementation of a Key Agreement algorithm).

In element 4.2, the Security Target author should specify the TSF data for which integrity validation is not required, and also specify the administrative role that is able to invoke the integrity verification process. While some TSF data are dynamic and therefore not amenable to integrity verification “makes sense” be subject to this requirement.

5.1.7.11 Cryptographic Self-Test (FPT_TST_(EXP).5)

FPT_TST_(EXP).5.1

The TSF shall run the suite of self-test [provided by the FIPS 140-2 cryptographic module during initial start-up (power on), at the request of the cryptographic administrator, periodically at an Administrator-specified interval not less than at least once a day] to demonstrate the correct operation of [the cryptographic components of the TSF].

FPT_TST_(EXP).5.2

The TSF shall be able to run the suite of self-tests provided by the FIPS 140-2 cryptographic module immediately after the generation of a key.

Application Note: For element 5.2, the Administrator has the ability to enable and disable this capability; this is specified in FMT_MOF.1.

5.1.8 Resource Utilization (FRU) Requirements

5.1.8.1 Full Priority of Service (FRU_PRS.2)

FRU_PRS.2.1

The TSF shall assign a priority to each subject in the TSF.

FRU_PRS.2.2

The TSF shall ensure that each access to all shareable resources shall be mediated on the basis of the subject's assigned priority.

Application Note: Certain TSF functions are typically more important than other functions. This requirement allows the TOE to assign resources to the subjects based on priority.

5.1.8.2 Maximum Quotas (FRU_RSA.1)

FRU_RSA.1.1

Refinement: The TSF shall enforce **Administrator-specified** maximum quotas on the following resources: [memory, processing power, data storage, [assignment: *controlled connection-oriented resources*]] that **users associated with** [*Administrator-specified network identifiers*] can use [over an **Administrator-specified** period of time].

Application Note: This requirement applies to a network entity attempting to exhaust the specified connection-oriented resources (or set of such resources) on the TOE.

The Security Target author should fill in the first assignment with the list of connection-oriented resources to which this requirement applies. That is, when a network entity uses such a connection-oriented resource, the TOE tracks that use for the purpose of determining whether the entity has exceeded the quota established by the Administrator.

5.1.9 TOE Access (FTA) Requirements

5.1.9.1 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

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

Application Note: The access banner applies whenever the TOE will provide a prompt for identification and authentication (e.g., administrators, authenticated proxy users). The intent of this requirement is to advise users of warnings regarding the unauthorized use of the TOE and to provide the Administrator with control over what is displayed (e.g. if the Security Administrator chooses, they can remove banner information that informs the user of the product and version number.

5.1.9.2 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1

Upon successful session establishment, the TSF shall display the [date and time] of the last successful session establishment to the user.

FTA_TAH.1.2

Upon successful session establishment, the TSF shall display the [date and time] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

5.1.9.3 TOE Session Establishment (FTA_TSE.1)

FTA_TSE.1.1

The TSF shall be able to deny session establishment based on [user location, time of day, day of week, and user role].

5.1.10 Trusted Path/Channels (FTP) Requirements

5.1.10.1 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

The TSF shall provide a communication path between itself and [*remote and local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2

The TSF shall permit [*the TSF, local users and remote users*] to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for [*initial user authentication, all writes to points in the TSF, all management of the TSF, [assignment: other services for which the trusted path is required]*].

5.2 Assurance Requirements

The assurance requirements in this Protection Profile are from the NIAP guidance for Medium Robustness Environments^{MRBT}. 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 explicit requirements have been created for some families in the ADV class both to remove ambiguity in the existing ADV requirements as well as to provide greater assurance than that associated with EAL4.

The set of assurance components are noted in Table 8. Those labeled with an EXP suffix are further described in various instructions in this document.

Table 8 – Assurance Requirements

Assurance Class	Assurance Components	
Configuration management	ACM_AUT.1	Partial CM Automation
	ACM_CAP.4	Generation Support and Acceptance Procedures
	ACM_SCP.2	Problem Tracking CM Coverage
Delivery and operation	ADO_DEL.2	Detection of Modification
	ADO_IGS.1	Installation, Generation and Start-up Procedures
Development	ADV_ARC_(EXP).1	Architectural Design
	ADV_FSP_(EXP).1	Functional Specification with Complete Summary
	ADV_HLD_(EXP).1	Security-Enforcing High-Level Design
	ADV_INT_(EXP).1	Modular Decomposition
	ADV_IMP.1	Subset of the Implementation of the TSF
	ADV_LLD_(EXP).1	Security-Enforcing Low-Level Design
	ADV_RCR.1	Informal Correspondence Demonstration
	ADV_SPM.1	Informal TOE Security Policy Model
	Guidance documents	AGD_ADM.1
AGD_USR.1		User Guidance
Life cycle support	ALC_DVS.1	Development Security
	ALC_FLR.2	Flaw Reporting Procedures
	ALC_LCD.1	Developer Defined Life-Cycle
	ALC_TAT.1	Tools and Techniques
Tests	ATE_COV.2	Analysis of Coverage
	ATE_DPT.2	Testing: Low-level Design
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent Testing-Sample
Vulnerability assessments	AVA_CCA_(EXP).2	Systematic Cryptographic Module Covert Channel Analysis
	AVA_MSU.2	Validation of Analysis
	AVA_SOF.1	Strength of TOE Security Function Evaluation
	AVA_VLA.3	Moderately Resistant

5 **5.2.1 Partial CM Automation (ACM_AUT.1)**

Developer action elements:

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

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

10 Content and presentation of evidence elements:

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

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

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

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

20 Evaluator action elements:

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

5.2.2 Generation Support And Acceptance Procedures (ACM_CAP.4)

Developer action elements:

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

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

ACM_CAP.4.3D – The developer shall provide CM documentation.

Content and presentation of evidence elements:

30 ACM_CAP.4.1C – The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C – The TOE shall be labeled with its reference.

ACM_CAP.4.3C – The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

35 ACM_CAP.4.4C – The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C – The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C – The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.

40 ACM_CAP.4.7C – The CM shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.8C – The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C – The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

45 ACM_CAP.4.10C – The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C – The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.4.12C – The CM system shall support the generation of the TOE.

- 50 ACM_CAP.4.13C – The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

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

5.2.3 Problem Tracking CM Coverage (ACM_SCP.2)

Developer action elements:

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

- 60 Content and presentation of evidence elements:

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

- 65 Evaluator action elements:

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

5.2.4 Detection of Modification (ADO_DEL.2)

Developer action elements:

- 70 ADO_DEL.2.1D – The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D – The developer shall use the delivery procedures.

Content and presentation of evidence elements:

- 75 ADO_DEL.2.1C – The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C – The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

- 80 ADO_DEL.2.3C – The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Evaluator action elements:

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

5.2.5 Installation, Generation, And Start-Up Procedures (ADO_IGS.1)

Developer action elements:

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

Content and presentation of evidence elements:

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

Evaluator action elements:

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

100 ADO_IGS.1.2E – The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.6 Architectural Design (ADV_ARC_(EXP).1)

Developer action elements:

105 ADV_ARC_(EXP).1.1D – The developer shall provide the architectural design of the TSF.

Content and presentation of evidence elements:

ADV_ARC_(EXP).1.1C – The presentation of the architectural design of the TSF shall be informal.

ADV_ARC_(EXP).1.2C – The architectural design shall be internally consistent.

110 ADV_ARC_(EXP).1.3C – The architectural design shall describe the design of the TSF self-protection mechanisms.

ADV_ARC_(EXP).1.4C – The architectural design shall describe the design of the TSF in detail sufficient to determine that the security enforcing mechanisms cannot be bypassed.

115 ADV_ARC_(EXP).1.5C – The architectural design shall justify that the design of the TSF achieves the self-protection function.

Evaluator Action Elements:

ADV_ARC_(EXP).1.1E – The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

120 ADV_ARC_(EXP).1.2E – The evaluator shall analyze the architectural design and dependent documentation to determine that FPT_SEP and FPT_RVM are accurately implemented in the TSF.

5.2.7 Functional Specification With Complete Summary (ADV_FSP_(EXP).1)

Developer Action Elements:

125 ADV_FSP_(EXP).1.1D
The developer shall provide a functional specification.

Content and Presentation of Evidence:

ADV_FSP_(EXP).1.1C – The functional specification shall completely represent the TSF.

130 ADV_FSP_(EXP).1.2C – The functional specification shall be internally consistent.

ADV_FSP_(EXP).1.3C – The functional specification shall describe the external TSF interfaces (TSFI's) using an informal style.

ADV_FSP_(EXP).1.4C – The functional specification shall designate each external TSFI as security enforcing or security supporting.

135 ADV_FSP_(EXP).1.5C – The functional specification shall describe the purpose and method of use for each external TSFI.

ADV_FSP_(EXP).1.6C – The functional specification shall identify and describe all parameters associated with each external TSFI.

140 ADV_FSP_(EXP).1.7C – For security enforcing external TSFI's, the functional specification shall describe the security enforcing effects and security enforcing exceptions.

ADV_FSP_(EXP).1.8C – For security enforcing external TSFI's, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions.

145

Evaluator action elements:

ADV_FSP_(EXP).1.1E – The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidenc.

150 ADV_FSP_(EXP).1.2E – The evaluator shall determine that the functional specification is an accurate and complete instantiation of the user visible TOE security functional requirements.

5.2.8 Security-Enforcing High-Level Design (ADV_HLD_(EXP).1)

Developer action elements:

155 ADV_HLD_(EXP).1.1C – The developer shall describe the structure of the TOE in terms of subsystems.

Content and presentation of evidence:

ADV_HLD_(EXP).1.1C – The high-level design shall describe the structure of the TOE in terms of subsystems.

160 ADV_HLD_(EXP).1.2C – The high-level design shall be internally consistent.

ADV_HLD_(EXP).1.3C – The high-level design shall describe the subsystems using an informal syle.

ADV_HLD_(EXP).1.4C – The high-level design shall describe the design of the TOE in sufficient detail to determine what subsystems of the TOE are part of the TSF.

165 ADV_HLD_(EXP).1.5C – The high-level design shall identify all subsystems in the TSF, and designate them as either security-enforcing or security-supporting.

ADV_HLD_(EXP).1.6C – The high-level design shall describe the structure of the security-enforcing subsystems.

170 ADV_HLD_(EXP).1.7C – For security-enforcing subsystems, the high-level design shall describe the design of the security-enforcing behavior.

ADV_HLD_(EXP).1.8C – For security-enforcing subsystems, the high-level design shall summarize any non-security-enforcing behavior.

ADV_HLD_(EXP).1.9C – The high-level design shall summarize the behavior for security-supporting subsystems.

175 ADV_HLD_(EXP).1.10C – The high-level design shall summarize all other interactions between subsystems of the TSF.

ADV_HLD_(EXP).1.11C – The high-level design shall describe any interactions between the security-enforcing subsystems of the TSF.

180 Evaluator action elements:

ADV_HLD_(EXP).1.1E – The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

185 ADV_HLD_(EXP).1.2E – The evaluator shall determine that the high-level design is an accurate and complete instantiation of all user-visible TOE security functional requirements with the exception of FPT_SEP and FPT_RVM.

5.2.9 Modular Decomposition (ADV_INT_(EXP).1)

Developer action elements:

ADV_INT_(EXP).1.1D – The developer shall design and implement the TSF using modular decomposition.

190 ADV_INT_(EXP).1.2D – The developer shall use sound software engineering principles to achieve the modular decomposition of the TSF.

ADV_INT_(EXP).1.3D – The developer shall design the modules such that they exhibit good internal structure and are not overly complex.

195 ADV_INT_(EXP).1.4D – The developer shall design modules that implement the [assignment: *list of SFP's*] such that they exhibit only functional, sequential, communicational, or temporal cohesion, with limited exceptions.

ADV_INT_(EXP).1.5D – The developer shall design the SFP-enforcing modules such that they exhibit only call or common coupling, with limited exceptions.

200 *Application Note: SFP enforcing modules are TSF modules that implement a specific SFP identified in ADV_INT_(EXP).1.4D.*

ADV_INT_(EXP).1.6D – The developer shall implement TSF modules using coding standards that result in good internal structure that is not overly complex.

205 ADV_INT_(EXP).1.7D – The developer shall provide a software architectural description.

Content and presentation of evidence elements:

ADV_INT_(EXP).1.1C – The software architectural description shall identify the SFP-enforcing and non-SFP-enforcing modules.

210 ADV_INT_(EXP).1.2C – The TSF modules shall be identical to those described by the low level design (ADV_LLD_(EXP).1.4C).

ADV_INT_(EXP).1.3C – The software architectural description shall provide a justification for the designation of non-SFP-enforcing modules that interact with the SFP-enforcing module(s).

215 ADV_INT_(EXP).1.4C – The software architectural description shall describe the process used for modular decomposition.

ADV_INT_(EXP).1.5C – The software architectural description shall describe how the TSF design is a reflection of the modular decomposition process.

220 ADV_INT_(EXP).1.6C – The software architectural description shall include the coding standards used in the development of the TSF.

ADV_INT_(EXP).1.7C – The software architectural description shall provide a justification, on a per module basis, of any deviations from the coding standards.

ADV_INT_(EXP).1.8C – The software architectural description shall include a coupling analysis that describes inter-module coupling for the SFP-enforcing modules.

225 ADV_INT_(EXP).1.9C – The software architectural description shall provide a justification, on a per module basis, for any coupling or cohesion exhibited by SFP-enforcing modules, other than those permitted.

230 ADV_INT_(EXP).1.10C – The software architectural description shall provide a justification, on a per module basis, that the SFP-enforcing modules are not overly complex.

Evaluator action elements

ADV_INT_(EXP).1.1E – The evaluator shall confirm that the information provided meets all the requirements for content and presentation of evidence.

235 ADV_INT_(EXP).1.2E – The evaluator shall perform a cohesion analysis for the modules that substantiates the type of cohesion claimed for a subset of SFP-enforcing modules.

ADV_INT_(EXP).1.3E – The evaluator shall perform a complexity analysis for a subset of TSF modules.

5.2.10 Subset Of The Implementation Of The TSF (ADV_IMP.1)

240 Developer action elements:

ADV_IMP.1.1D – The developer shall provide the implementation representation for a selected subset of the TSF.

Content and presentation of evidence elements:

245 ADV_IMP.1.1C – The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C – The implementation representation shall be internally consistent.

Evaluator action elements:

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

ADV_IMP.1.2E – The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

255 5.2.11 Security-Enforcing Low-Level Design (ADV_LLD_(EXP).1)

Developer action elements:

ADV_LLD_(EXP).1.1D – The developer shall provide the low-level design of the TSF.

260 Content and presentation of evidence elements:

ADV_LLD_(EXP).1.1C – The presentation of the low-level design shall be informal.

ADV_LLD_(EXP).1.2C – The presentation of the low-level design shall be separate from the implementation representation.

ADV_LLD_(EXP).1.3C – The low-level design shall be internally consistent.

265 ADV_LLD_(EXP).1.4C – The low-level design shall identify and describe data that are common to more than one module, where any of the modules is a security-enforcing module.

ADV_LLD_(EXP).1.5C – The low-level design shall describe the TSF in terms of modules, designating each module as either security-enforcing or security-supporting.

270 ADV_LLD_(EXP).1.6C – The low-level design shall describe each security-enforcing module in terms of its purpose, interfaces, return values from those interfaces, called interfaces to other modules, and global variables.

ADV_LLD_(EXP).1.7C – For each security-enforcing module, the low-level design shall provide an algorithmic description detailed enough to represent the TSF implementation.

275

Application Note: An algorithmic description contains sufficient detail such that two different programmers would produce functionally-equivalent code, although data structures, programming methods, etc. may differ.

280 ADV_LLD_(EXP).1.8C – The low-level design shall describe each security-supporting module in terms of its purpose and interaction with other modules.

Evaluator action elements:

285 ADV_LLD_(EXP).1.1E – The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD_(EXP).1.2E – The evaluator shall determine that the low-level design is an accurate and complete instantiation of all TOE security functional requirements, with the exception of FPT_SEP and FPT_RVM.

290 **5.2.12 Informal Correspondence Demonstration (ADV_RCR.1)**

Developer action elements:

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

295 Content and presentation of evidence elements:

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

300 Evaluator action elements:

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

5.2.13 Informal TOE Security Policy Model (ADV_SPM.1)

Developer action elements:

305 ADV_SPM.1.1D – The developer shall provide a TSP model.

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

Content and presentation of evidence elements:

310 ADV_SPM.1.1C – The TSP model shall be informal.

ADV_SPM.1.2C – The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C – The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

315 ADV_SPM.1.4C – The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

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

5.2.14 Administrator Guidance (AGD_ADM.1)

Developer action elements:

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

Content and presentation of evidence elements:

AGD_ADM.1.1C – The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

330 AGD_ADM.1.2C – The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C – The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

335 AGD_ADM.1.4C – The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C – The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

340 AGD_ADM.1.6C – The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed., including changing the security characteristics of entities under control of the TSF.

AGD_ADM.1.7C – The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C – The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

345

Evaluator action elements:

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

5.2.15 User Guidance (AGD_USR.1)

350 Developer action elements:

AGD_USR.1.1D – The developer shall provide user guidance.

Content and presentation of evidence elements:

355 AGD_USR.1.1C – The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C – The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C – The user guidance shall contain warnings about user-accessible functions and privilege that should be controlled in a secure processing environment.

360 AGD_USR.1.4C – The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of the TOE security environment.

AGD_USR.1.5C – The user guidance shall be consistent with all other documentation supplied for evaluation.

365 AGD_USR.1.6C – The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

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

5.2.16 Development Security (ALC_DVS.1)

Developer action elements:

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

375 Content and presentation of evidence elements:

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

380 ALC_DVS.1.2C – The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

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

5.2.17 Flaw Reporting Procedures (ALC_FLR.2)

Developer action elements:

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

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

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

395 Content and presentation of evidence 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.

400 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.

405 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 user's reports and enquires of suspected security flaws in the TOE.

410 ALC_FLR.2.6C – The procedures for processing reported security flaw shall ensure that any reported flaws are corrected and the correction 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.

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.

415

Evaluator action elements:

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

420 **5.2.18 Developer Defined Life-Cycle (ALC_LCD.1)**

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.

425 ALC_LCD.1.2D – The developer shall provide life-cycle definition documentation.

Content and presentation of evidence elements:

ALC_LCD.1.1C – The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

430 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 life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

435 **5.2.19 Tools and Techniques (ALC_TAT.1)**

Developer action elements:

ALC_TAT.1.1D – The developer shall identify the development tools being used for the TOE.

440 ALC_TAT.1.2D – The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C – All development tools used for implementation shall be well-defined.

445 ALC_TAT.1.2C – The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C – The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

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

5.2.20 Analysis Of Coverage (ATE_COV.2)

Developer action elements:

455 ATE_COV.2.1D – The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements;

ATE_COV.2.1C – The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

460 ATE_COV.2.2C – The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

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

5.2.21 Testing: Low-Level Design (ATE_DPT.2)

Developer action elements:

470 ATE_DPT.2.1D – The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.2.1C – The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.

475

Evaluator action elements:

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

5.2.22 Functional Testing (ATE_FUN.1)

480 Developer action elements:

ATE_FUN.1.1D – The developer shall test the TSF and document the results.

ATE_FUN.1.2D – The developer shall provide test documentation.

Content and presentation of evidence elements:

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

ATE_FUN.1.2C – The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

490 ATE_FUN.1.3C – The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C – The expected test results shall show the anticipated outputs from a successful execution of the tests.

495 ATE_FUN.1.5C – The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

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

500

5.2.23 Independent Testing-Sample (ATE_IND.2)

Developer action elements:

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

505 Content and presentation of evidence elements:

ATE_IND.2.1C – The TOE shall be suitable for testing.

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

510 Evaluator action elements:

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

ATE_IND.2.2E – The evaluator shall test a subset of the TSF as appropriate to confirm that the TPE operates as specified.

515 ATE_IND.2.3E – The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

**5.2.24 Systematic Cryptographic Module Covert Channel Analysis
(AVA_CCA_(EXP).2)**

Developer Action Elements:

520 AVA_CCA_(EXP).2.1D – 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.

AVA_CCA_(EXP).2.2D – The developer shall provide covert channel analysis documentation.

525

Application Note: 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.

530

535 Content and Presentation of Evidence Elements:

AVA_CCA_(EXP).2.1C – The analysis documentation shall identify covert channels in the cryptographic module and estimate their capacity.

540 AVA_CCA_(EXP).2.2C – 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_(EXP).2.3C – The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA_CCA_(EXP).2.4C – The analysis documentation shall describe the method used for estimating channel capacity, based on worst-case scenarios.

545 AVA_CCA_(EXP).2.5C – The analysis documentation shall describe the worst-case exploitation scenario for each identified covert channel.

AVA_CCA_(EXP).2.6C – The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

550 Evaluator Action Elements:

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

AVA_CCA_(EXP).2.2E – The evaluator shall confirm the results of the covert channel analysis show that the cryptographic module meets its functional requirements.

555 AVA_CCA_(EXP).2.3E – The 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.

560 **5.2.25 Validation Of Analysis (AVA_MSU.2)**

Developer action elements:

AVA_MSU.2.1D – The developer shall provide guidance documentation.

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

565

Content and presentation of evidence elements:

AVA_MSU.2.1C – The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

570 AVA_MSU.2.2C – The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C – The guidance documentation shall list all assumptions about the intended environment.

575 AVA_MSU.2.4C – The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C – The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

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

5.2.26 Stength Of TOE Security Function Evaluation (AVA_SOF.1)

Developer action elements:

585 AVA_SOF.1.1D – The develop shall perform a strength of TOE security function analysis for each mechanism identified in the Security Target as having a strength of TOE security function claim.

Content and presentation of evidence elements:

590 AVA_SOF.1.1C – For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the Protection Profile / Security Target.

595 AVA_SOF.1.2C – For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the Protection Profile / Security Target.

Evaluator action elements:

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

600 AVA_SOF.1.2E – The evaluator shall confirm that the strength claims are correct.

5.2.27 Moderately Resistant (AVA_VLA.3)

Developer action elements:

AVA_VLA.3.1D – The developer shall perform a vulnerability analysis.

AVA_VLA.3.2D – The developer shall provide vulnerability analysis documentation.

605

Content and presentation of evidence elements:

AVA_VLA.3.1C – The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

610

AVA_VLA.3.2C – The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.3.3C – The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

615

AVA_VLA.3.4C – The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.3.5C – The vulnerability analysis documentation shall show that the search for vulnerabilities is systemic.

Evaluator action elements:

620

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

AVA_VLA.3.2E – The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

625

AVA_VLA.3.3E – The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.3.4E – The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

630

AVA_VLA.3.5E – The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.

6 Rationale

6.1 Rationale for TOE Security Objectives

Table 9 – Objectives Mapped to Threats and Policies

635

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.ADMIN_ERROR</p> <p>An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.</p>	<p>O.ROBUST_ADMIN_GUIDANCE</p> <p>The TOE will provide administrators with the necessary information for secure delivery and management.</p>	<p>O.ROBUST_ADMIN_GUIDANCE (ADO_DEL.2, ADO_IGS.1, AGD_ADM.1, AGD_USR.1, AVA_MSU.2) help to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process.</p> <p>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.</p>
	<p>O.ADMIN_ROLE</p> <p>The TOE will provide administrator role to isolate administrative actions, and to make the administrative functions available locally and remotely.</p>	<p>O.ADMIN_ROLE (FDP_ACC.2, FMT_SMR.2) plays a role in mitigating this threat by limiting who can perform administrative functions that would affect the security of the TOE.</p> <p>For example, only users in the administrator role would be allowed to add users, change security event log settings, and modify security parameters.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.MANAGE</p> <p>The TOE will provide all the functions and facilities necessary to support the administrators in their management of security of the TOE, and restrict these functions and facilities from unauthorized use.</p>	<p>O.MANAGE (FDP_ACF.1-NIAP-0407, FDP_ETC.2, FDP_ITC.2, FDP_ROL.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_REV.1, FMT_SMF.1) also contributes to mitigating this threat by providing administrators the capability to view and modify configuration settings.</p>
<p>T.ADMIN_ROGUE</p> <p>An administrator's intentions may become malicious resulting in user or TSF data being compromised.</p>	<p>O.AUDIT_GENERATION</p> <p>The TOE will provide the capability to detect and create records of security-relevant events.</p>	<p>O.AUDIT_GENERATION (FAU_GEN.1-NIAP-0407, FAU_GEN.2-NIAP-0410, FAU_SEL.1-NIAP-0407, FIA_USB.1, FTA_TAH.1) creates a record that can be used to detect rogue administrator's malicious acts.</p>
	<p>O.AUDIT_PROTECTION</p> <p>The TOE will provide the capability to protect audit information.</p>	<p>O.AUDIT_PROTECTION (FAU_SAR.2, FAU_STG.2-NIAP-0429, FAU_STG.3, FAU_STG.4, FMT_MOF.1, FMT_MTD.1, FMT_MTD.2) prevents a user in the administrator role from modifying the audit records. It will not prevent an administrator from deleting an audit record.</p>
	<p>O.AUDIT_REVIEW</p> <p>The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.</p>	<p>O.AUDIT_REVIEW (FAU_ARP.1, FAU_SAA.1-NIAP-0407, FAU_SAR.1) helps to identify an administrator taking malicious actions. This is more effective if there is more than one administrator that receive automated alerts.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.AUDIT_COMPROMISE</p> <p>A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's actions.</p>	<p>O.AUDIT_PROTECTION</p> <p>The TOE will provide the capability to protect audit information.</p>	<p>O.AUDIT_PROTECTION (FAU_SAR.2, FAU_STG.2-NIAP-0429, FAU_STG.3, FAU_STG.4, FMT_MOF.1, FMT_MTD.1, FMT_MTD.2) contributes to mitigating this threat by controlling access to the audit trail. Users in the Administrator role and any trusted IT entities are the only ones allowed to read the audit trail.</p> <p>No user or system is allowed to modify audit records, and only users in the Administrator role are allowed to delete audit records in the audit trail.</p> <p>The TOE has the capability of notifying Administrators if the audit trail is approaching its capacity.</p>
	<p>O.ADMIN_ROLE</p> <p>The TOE will provide administrator role to isolate administrative actions, and to make the administrative functions available locally and remotely.</p>	<p>O.ADMIN_ROLE (FDP_ACC.2, FMT_SMR.2) plays a role in mitigating this threat by limiting deletion of audit records to users in the Administrator role.</p>
	<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) prevents a user not authorized to read the audit trail from accessing audit information that might otherwise be persistent in a TOE resource, such as in memory.</p> <p>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.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.SELF_PROTECTION</p> <p>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.</p>	<p>O.SELF_PROTECTION (FDP_SDI.2, FPT_FLS.1, FPT_ITI.1, FPT_RVM.1, FPT_PHP.2, FPT_SEP.2) contributes to countering this threat by ensuring 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 trail. Likewise, ensuring that the functions that protect the audit trail are always invoked is also critical to the mitigation of this threat.</p>
<p>T.CRYPTO_COMPROMISE</p> <p>A malicious user or process may cause key, data or executable code associate with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and data protected by those mechanisms.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) 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.</p>
	<p>O.SELF_PROTECTION</p> <p>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.</p>	<p>O.SELF_PROTECTION (FDP_SDI.2, FPT_FLS.1, FPT_ITI.1, FPT_RVM.1, FPT_PHP.2, FPT_SEP.2) contributes to countering this threat by ensuring 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 resources under its control, which includes the cryptographic data and executable code.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.DOCUMENT_KEY_LEAKAGE</p> <p>The bandwidth of channels that can be used to compromise key material shall be documented.</p>	<p>O.DOCUMENT_KEY_LEAKAGE (AVA_CCA_(EXP).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 specific time period.</p>
	<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) addresses this threat by preventing access to crypto information, such as crypto keys, states, or initialization vectors, after it is no longer needed.</p>
<p>T.FLAWED_DESIGN</p> <p>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.</p>	<p>O.CHANGE_MANAGEMENT</p> <p>The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.</p>	<p>O.CHANGE_MANAGEMENT (ACM_AUT.1, ACM_CAP.4, ACM_SCP.2, 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.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.SOUND_DESIGN</p> <p>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.</p>	<p>O.SOUND_DESIGN (ADV_FSP_(EXP).1, ADV_HLD_(EXP).1, ADV_INT_(EXP).1, ADV_LLD_(EXP).1, ADV_ARC_(EXP).1, ADV_RCR.1, ADV_SPM.) 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.</p>
	<p>O.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>O.VULNERABILITY_ANALYSIS_TEST (AVA_VLA.3) 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.</p>
<p>T.FLAWED_IMPLEMENTATION</p> <p>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.</p>	<p>O.CHANGE_MANAGEMENT</p> <p>The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.</p>	<p>O.CHANGE_MANAGEMENT (ACM_AUT.1, ACM_CAP.4, ACM_SCP.2, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1) 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.</p>

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.SOUND_IMPLEMENTATION</p> <p>The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.</p>	<p>O.SOUND_IMPLEMENTATION (ADV_IMP.2, ADV_INT_(EXP).1, ADV_LLD_(EXP).1, ADV_ARC_(EXP).1, ADV_RCR.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.</p>
	<p>O.THOROUGH_FUNCTIONAL_TESTING</p> <p>The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.</p>	<p>O.THOROUGH_FUNCTIONAL_TESTING (ATE_COV.2, ATE_FUN.1, ATE_IND.2, ATE_DPT.2) increases the likelihood that any errors that have been introduced despite the previous two objectives are discovered through testing.</p>
	<p>O.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>O.VULNERABILITY_ANALYSIS_TEST (AVA_VLA.3) 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 the 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.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.MASQUERADE</p> <p>A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.</p>	<p>O.ROBUST_TOE_ACCESS</p> <p>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.</p>	<p>O.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2, FTA_TSE.1, AVA_SOF.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 an attacker attempting to login and masquerade as an authorized user.</p>
<p>T.MALICIOUS_TSF_COMPROMISE</p> <p>A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified or deleted).</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) 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.</p>
	<p>O.SELF_PROTECTION</p> <p>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.</p>	<p>O.SELF_PROTECTION (FDP_SDI.2, FPT_FLS.1, FPT_ITI.1, FPT_RVM.1, FPT_PHP.2, FPT_SEP.2) 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.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.MANAGE</p> <p>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.</p>	<p>O.MANAGE (FDP_ACF.1-NIAP-0407, FDP_ETC.2, FDP_ITC.2, FDP_ROL.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_REV.1, FMT_SMF.1) provides the capability to restrict access to TSF to those that are authorized to use the functions. Satisfaction of this objective prevents unauthorized access to TSF functions and data through the administrative mechanisms.</p>
	<p>O.TRUSTED_PATH</p> <p>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.</p>	<p>O.TRUSTED_PATH (FTP_TRP.1) plays a role in addressing this threat by ensuring that there is a trusted communication path between the TSF and various users and trusted IT entities. This ensures the transmitted data cannot be compromised or disclosed during the duration of the trusted path.</p>
<p>T.POOR_TEST</p> <p>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.</p>	<p>O.CORRECT_TSF_OPERATION</p> <p>The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.</p>	<p>O.CORRECT_TSF_OPERATION (FPT_AMT.1, FPT_TST_(EXP).4, FPT_TST_(EXP).5, FPT_RVM.1) 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.</p>

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.THOROUGH_FUNCTIONAL_TESTING</p> <p>The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.</p>	<p>O.THOROUGH_FUNCTIONAL_TESTING (ATE_COV.2, ATE_FUN.1, ATE_IND.2, ATE_DPT.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 TSF cannot be used in unintended ways to circumvent the TOE's security policies.</p>
	<p>O.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>O.VULNERABILITY_ANALYSIS_TEST (AVA_VLA.3) 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.</p>
<p>T.REPLAY</p> <p>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 transmitted during the course of legitimate use).</p>	<p>O.REPLAY_DETECTION</p> <p>The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.</p>	<p>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 and then replay it to gain access to the TOE, as well as counter the ability of a user to act as another user.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.RESOURCE_EXHAUSTION</p> <p>A malicious process or user may block others from system resources (e.g., flooding the TOE with poll requests) via a resource exhaustion denial of service attack.</p>	<p>O.RESOURCE_SHARING</p> <p>The TOE shall provide mechanisms that mitigate attempts to exhaust the memory, computing and input/output resources provided by the TOE.</p>	<p>O.RESOURCE_SHARING (FMT_MOF.1, FMT_MTD.2, FRU_PRS.2, FRU_RSA.1) mitigates this threat by requiring the TOE to provide controls relating to three different resources: CPU time, memory allocation and input/output bandwidth usage.</p> <p>The administrator is allowed to specify a percentage of processor time, maximum amount of RAM, and maximum amount of input/output bandwidth that is allowed to be used by any user or system communication with the TOE.</p> <p>The objective addresses the denial-of-service attack of a user attempting to exhaust the TOE resources.</p>
<p>T.SPOOFING</p> <p>A malicious user, process, or external entity may misrepresent itself as the TOE to obtain identification and authentication data.</p>	<p>O.TRUSTED_PATH</p> <p>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.</p>	<p>O.TRUSTED_PATH (FTP_TRP.1) mitigates this threat by ensuring users have the capability to ensure they are communicating with the TOE when providing identification and authentication data to the TOE.</p> <p>This trusted path prevents man-in-the-middle attacks.</p>
<p>T.UNATTENDED_SESSION</p> <p>A user may gain unauthorized access to an unattended session.</p>	<p>O.ROBUST_TOE_ACCESS</p> <p>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.</p>	<p>O.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2, FTA_TSE.1, AVA_SOF.1) helps to mitigate this threat by including by including mechanisms the place controls on user's sessions.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.UNAUTHORIZED_ACCESS</p> <p>A user may gain access to user data for which they are not authorized according to the TOE security policy.</p>	<p>O.MEDIATE</p> <p>The TOE must protect user data in accordance with its security policy.</p>	<p>O.MEDIATE (FDP_ACC.2, FDP_ACF.1-NIAP-0407, FDP_IFC.2, FDP_IFF.1) works to mitigate this threat by requiring objects in the directory are protected using access control items. An access control item contains information about who is allowed to access an object, as well as the allowed modes of access. The settings present in the access control item selected in the access control decision process determine whether or not a user is authorized to access the object. It is required that all objects be covered by an access control item. Note that O.SELF_PROTECTION ensures that this access control mechanism is always invoked.</p>
	<p>O.USER_GUIDANCE</p> <p>The TOE will provide users with the information necessary to correctly use the security mechanisms.</p>	<p>O.USER_GUIDANCE (AGD_USR.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.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>T.UNIDENTIFIED_ACTIONS</p> <p>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.</p>	<p>O.AUDIT_REVIEW</p> <p>The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.</p>	<p>O.AUDIT_REVIEW (FAU_ARP.1, FAU_SAA.1-NIAP-0407, FAU_SAR.1) mitigates 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.</p> <p>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 and immediately notifies an administrator once an event has occurred or a set threshold has been met.</p> <p>The TOE also has the capability to export the audit information to an external audit analysis tool (such as a security event monitoring (SEM) product or managed security service) for more detailed or composite audit analysis.</p>
<p>T.UNKNOWN_STATE</p> <p>When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.</p>	<p>O.MAINT_MODE</p> <p>The TOE shall provide a mode from which recovery or initial startup procedures can be performed.</p>	<p>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.</p>

Threat / Policy	Objectives Addressing Threat	Rationale
	<p>O.CORRECT_TSF_OPERATION</p> <p>The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.</p>	<p>O.CORRECT_TSF_OPERATION (FPT_AMT.1, FPT_TST_(EXP).4, FPT_TST_(EXP).5, FPT_RVM.1) 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.</p>
	<p>O.SOUND_DESIGN</p> <p>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.</p>	<p>O.SOUND_DESIGN (ADV_FSP_(EXP).1, ADV_HLD_(EXP).1, ADV_INT_(EXP).1, ADV_LLD_(EXP).1, ADV_ARC_(EXP).1, ADV_RCR.1, ADV_SPM.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.</p>
	<p>O.ROBUST_ADMIN_GUIDANCE</p> <p>The TOE will provide administrators with the necessary information for secure delivery and management.</p>	<p>O.ROBUST_ADMIN_GUIDANCE (ADO_DEL.2, ADO_IGS.1, AGD_ADM.1, AGD_USR.1, AVA_MSU.2) 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 manner. The guidance also provides information about the corrective measure necessary when a failure occurs.</p>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>P.ACCESS_BANNER</p> <p>The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent to by accessing the TOE.</p>	<p>O.DISPLAY_BANNER</p> <p>The TOE will display an advisory warning regarding use of the TOE.</p>	<p>O.DISPLAY_BANNER (FTA_TAB.1) satisfies this policy by ensuring that the TOE displays an 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.</p>
<p>P.ACCOUNTABILITY</p> <p>The authorized users of the TOE shall be held accountable for their actions within the TOE.</p>	<p>O.AUDIT_GENERATION</p> <p>The TOE will provide the capability to detect and create records of security-relevant events.</p>	<p>O.AUDIT_GENERATION (FAU_GEN.1-NIAP-0407, FAU_GEN.2-NIAP-0410, FAU_SEL.1-NIAP-0407, FIA_USB.1, FTA_TAH.1) addresses this policy by providing an audit mechanism to record the actions of a specific user. The administrator's ID is recorded when any security relevant change is made to the TOE. Attributes used in the audit record generation process are also required to be bound to the subject, ensuring users are held accountable.</p>
	<p>O.TIME_STAMPS</p> <p>The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.</p>	<p>O.TIME_STAMPS (FMT_MTD.1, FPT_STM.1) plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (configured by a trusted NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID will also include the date and time that the event occurred.</p>
	<p>O.ROBUST_TOE_ACCESS</p> <p>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.</p>	<p>O.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2, FTA_TSE.1, AVA_SOF.1) 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>

Version 0.71 – May 18, 2006

Threat / Policy	Objectives Addressing Threat	Rationale
<p>P.ADMIN_ACCESS</p> <p>Administrators shall be able to administer the TOE both locally and remotely through protected communication channels.</p>	<p>O.ADMIN_ROLE</p> <p>The TOE will provide administrator role to isolate administrative actions, and to make the administrative functions available locally and remotely.</p>	<p>O.ADMIN_ROLE (FDP_ACC.2, FMT_SMR.2) supports this policy by requiring the TOE to provide mechanisms 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.</p>
	<p>O.TRUSTED_PATH</p> <p>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.</p>	<p>O.TRUSTED_PATH (FTP_TRP.1) satisfies this policy by requiring that each remote administrative and management session for all trusted users is authenticated and conducted via a secure channel. Additionally, all trusted IT entities (e.g., log collectors, SEM products) connect through a protected channel, thus avoiding disclosure and spoofing problems.</p>
<p>P.CRYPTOGRAPHY</p> <p>The TOE shall use NIST FIPS validated cryptography as a baseline with additional NSA approved methods for key management (i.e., generation, access, distribution, destruction, handling, and storage of keys) and cryptographic operations (i.e., encryption, decryption, signature, hashing, key exchange and random number generation services).</p>	<p>O.CRYPTOGRAPHY</p> <p>The TOE shall use NIST FIPS 140-2 validated cryptographic services.</p>	<p>O.CRYPTOGRAPHY (FCS_BCM_(EXP).1.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1) directly addresses this policy by restricting the cryptographic services to FIPS 14-2 validated services.</p>
	<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.</p>	<p>O.CRYPTO_RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) addresses this policy by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.</p>

Threat / Policy	Objectives Addressing Threat	Rationale
<p>P.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>O.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>O.VULNERABILITY_ANALYSIS_TEST (AVA_VLA.3) 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.</p>

6.2 Rationale for the Security Objectives and Security Requirements for the TOE

Table 10 – Requirements Mapped to Objectives

Objectives	Requirements Addressing Objectives	Rationale
<p>O.ADMIN_ROLE</p> <p>The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.</p>	<p>FDP_ACC.2</p> <p>FMT_SMR.2</p>	<p>FDP_ACC.2 requires the enforcement of an access control policy that isolates the administrative actions to users in the Administrator role.</p> <p>FMT_SMR.2 requires an Administrator role that is responsible for configuring security-relevant parameters on the TOE. The TSF is able to associate a human user with one of the three required roles: Administrator, Operator and Display.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.AUDIT_GENERATION</p> <p>The TOE will provide the capability to detect and create records of security-relevant events.</p>	<p>FAU-GEN.1-NIAP-0407 FAU_GEN.2-NIAP-0410 FAU_SEL.1-NIAP-0407 FIA_USB.1 FTA_TAH.1</p>	<p>FAU-GEN.1-NIAP-0407 defines the set of events the TOE must be capable of recording in the audit log. It also specifies the minimum information that must be available in each audit record. Table 5 lists all additional information required in an audit event that is related to a specific functional requirement. A refinement of this SFR requires an indicator in the audit record to identify if it is a security record or not. This requirement also places a requirement on the Security Target author to specify any additional audit events for any security functional requirements the author adds.</p> <p>FAU_GEN.2-NIAP-0410 ensures that the audit records associate a user identity with the auditable event.</p> <p>FAU_SEL.1-NIAP-0407 allows Administrators to configure which auditable events will be recorded in the audit trail. This provide the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.</p> <p>FIA_USB.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. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed.</p> <p>FTA_TAH.1 requires the logging of all TSF sessions.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.AUDIT_PROTECTION</p> <p>The TOE will provide the capability to protect audit information.</p>	<p>FAU_SAR.2 FAU_STG.2-NIAP-0429 FAU_STG.3 FAU_STG.4 FMT_MOF.1 FMT_MTD.1 FMT_MTD.2</p>	<p>FAU_SAR.2 restricts the ability to read the security event audit trail to Administrators, thus preventing the disclosure of security events 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 a file off of the TOE).</p> <p>The FAU_STG family dictates how the audit trail is protected. FAU_STG.2-NIAP-0429 restricts the ability to delete audit records to Administrators. No one, including Administrators, is allowed to modify the audit records. This prevents log forgery. Finally it requires one hour of audit records be retained even if the audit log fails or is full.</p> <p>FAU_STG.3 provides an alarm when the audit log reaches an Administrator configurable threshold. This allows the Administrator to manage the audit trail before it becomes full and avoiding the possible loss of audit data.</p> <p>FAU_STG.4 requires the TOE overwrite the oldest stored audit records. The TOE will always keep the most current audit records as audit space is available.</p> <p>FMT_MOF.1 restricts the ability to control the behavior of the audit mechanism to the Administrator role. The Administrator is the only role that controls the behavior of the events that generate alarms and whether the alarm mechanism is enabled or disabled.</p> <p>FMT_MTD.1 restricts the ability to change the defaults, modify and delete audit records to the Administrator role.</p> <p>FMT_MTD.2 restricts the limits to the audit records to the Administrator. The Administrator will be able to set the maximum audit record size. A larger size will reduce the likelihood of audit record loss due to the audit record space being full.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.AUDIT_REVIEW</p> <p>The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.</p>	<p>FAU_ARP.1 FAU_SAA.1-NIAP-0407 FAU_SAR.1</p>	<p>FAU_ARP.1 addresses this objective by requiring a real time security alarm be available for display on a HMI. This will facilitate an Operator and Administrator response to an audit event.</p> <p>FAU_SAA.1-NIAP-0407 defines the events that indicate a potential security violation and will generate an alarm. The triggers for these events are largely configurable by the Administrator.</p> <p>FAU_SAR.1 provides Administrators the capability to read all of the security related audit events.</p>
<p>O.CHANGE_MANAGEMENT</p> <p>The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.</p>	<p>ACM_AUT.1 ACM_CAP.4 ACM_SCP.2 ALC_DVS.1 ALC_FLR.2 ALC_LCD.1</p>	<p>ACM_AUT.1 complements ACM_CAP.4, 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.</p> <p>ACM_CAP.4 contributes to this objective by requiring the developer have a configuration management plan that describes how changes to the TOE and its evaluation deliverables are managed. The developer is also required to employ a configuration management system that operates in accordance with the CM plan and provides the capability to control who on the development staff can make changes to the TOE and its developed evidence. This requirement also ensures that authorized changes to the TOE have been analyzed and the developer's acceptance plan describes how this analysis is performed and how decisions to incorporate the changes to the TOE are made</p> <p>ACM_SCP.2 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.</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>ALC_DVS.1 requires the developer describe the security measures they employ to ensure the integrity and confidentiality of the TOE are maintained. The physical, procedural, and personnel security measures the developer uses provides an added level of control over who and how changes are made to the TOE and its associated evidence.</p> <p>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.</p> <p>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.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.CORRECT_TSF_OPERATION</p> <p>The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.</p>	<p>FPT_AMT.1 FPT_TST_(EXP).4 FPT_TST_(EXP).5 FPT_RVM.1</p>	<p>FMT_AMT.1 requires tests to demonstrate the proper operation of the abstract machine in the software portions of the TSF. The tests are run at start-up, periodically and on-demand to ensure correct operation of the abstract machine.</p> <p>FPT_TST_(EXP).4 has been created to ensure Administrator 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 been corrupted. Hardware failures could render a TOE's software ineffective in enforcing its security policies and this requirement provides the end user the ability to discover any failures in the hardware security mechanisms.</p> <p>FPT_TST_(EXP).5 is 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.</p> <p>FPT_RVM.1 prevents bypassing the security functions in the TSF.</p>
<p>O.CRYPTO_RESIDUAL_INFORMATION</p> <p>The TOE will ensure that any information associated with the cryptographic functionality and contained in protected resource is not released when the resource is reallocated.</p>	<p>FCS_CKM.2 FCS_CKM.4</p>	<p>FCS_CKM.2 requires protection of the crypto keys in transit per the FIPS 140-2 standard. This includes preventing disclosure of secret keys.</p> <p>FCS_CKM.4 addresses this objective by requiring secure destruction of crypto keys. The crypto keys will not be available as residual information.</p> <p>Note: This objective was modified from O.RESIDUAL_INFORMATION in the CCEVS instructions for a medium robustness environment. Information in the TOE is not highly confidential with the exception of the security information.</p>

Version 0.71 – May 18, 2006

Objectives	Requirements Addressing Objectives	Rationale
<p>O.CRYPTOGRAPHY</p> <p>The TOE shall use NIST FIPS 140-2 validated cryptographic services.</p>	<p>FCS_BCM_(EXP).1.1 FCS_CKM.1 FCS_CKM.2 FCS_CKM.4 FCS_COP.1</p>	<p>The entire FCS class or requirements in this Protection Profile require FIPS 140-2 validated services.</p> <p>FCS_BCM_(EXP).1.1 requires a FIPS 140-2 certified module, whether it is a hardware, software or combination module.</p> <p>FCS_CKM.1 requires FIPS 140-2 approved key generation algorithms and key sizes.</p> <p>FCS_CKM.2 requires a FIPS 140-2 approved key distribution methods.</p> <p>FCS_CKM.4 requires secure key destruction per the FIPS 140-2 standard.</p> <p>FCS_COP.1 requires all cryptographic functions be performed in a FIPS 140-2 manner.</p>
<p>O.DISPLAY_BANNER</p> <p>The TOE will display an advisory warning regarding use of the TOE.</p>	<p>FTA_TAB.1</p>	<p>FTA_TAB.1 meets this objective by requiring the TOE display an Administrator-specified banner before a user is allowed any access to the TOE. This banner includes warnings regarding unauthorized use of the TOE.</p>
<p>O.DOCUMENT_KEY_LEAKAGE</p> <p>The bandwidth of channels that can be used to compromise key material shall be documented.</p>	<p>AVA_CCA_(EXP).2</p>	<p>AVA_CCA_(EXP).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.</p>
<p>O.MAINT_MODE</p> <p>The TOE shall provide a mode from which recovery or initial startup procedures can be performed.</p>	<p>FPT_RCV.2</p>	<p>FPT_RCV.2 meets this objective by ensuring the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. Upon 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 follow documented procedures that instruct them on to return to the TOE to a secure state.</p>
<p>O.MANAGE</p> <p>The TOE will provide all the</p>	<p>FDP_ACF.1-NIAP-0407 FDP_ETC.2 FDP_ITC.2</p>	<p>FDP_ACF.1-NIAP-0407 details the manner in which objects are to be protected. This</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>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.</p>	<p>FDP_ROL.1 FMT_MSA.1 FMT_MSA.3 FMT_MTD.1 FMT_MTD.2 FMT_MTD.3 FMT_REV.1 FMT_SMF.1</p>	<p>includes management functions and facilities.</p> <p>FDP_ETC.2 requires export of user data with security attributes be protected. Specifically this data must have a security field that can be used to detect unauthorized modification. This data may be exported for management purposes or to support the deployment of devices with a similar configuration.</p> <p>FDP_ITC.2 is the converse of FDP_ETC.2. FDP_ITC.2 verifies that imported user data with security attributes has not been altered and that it is imported properly.</p> <p>FDP_ROL.1 requires the TOE to be able to rollback the last three changes to the security configuration. If an Administrator makes a mistake, this requirement supports quick recovery.</p> <p>FMT_MSA.1 restricts the ability to ability to change defaults, modify or delete security attributes to the Administrator role. Operators and Display users, and unauthorized persons, will not be able to manage the security attributes in the TOE.</p> <p>FMT_MSA.3 restricts the ability to change the default security attributes to the Administrator role.</p> <p>FMT_MTD.1 restricts the ability to modify or delete TSF data to the Administrator role. This is data in audit logs or process data, as opposed to the configuration data that is covered in FMT_MSA.</p> <p>FMT_MTD.2 allows the Administrator, and only the Administrator, to configure parameters that will identify and stop denial of service attacks.</p> <p>FMT_MTD.3 prevents an Administrator from specifying values that are insecure. For example, this could prevent an Administrator from entering values that would result in a loss of availability. The Security Target will specify these values and limits.</p> <p>FMT_REV.1 allows the Administrator to revoke security attributes associated with users, subjects and objects. This could be used to remove authorization rights for a users or isolating points in a field device.</p>

Version 0.71 – May 18, 2006

Objectives	Requirements Addressing Objectives	Rationale
		<p>FMT_REV.1 also requires the TOE revoke these rights in an Administrator configurable time period.</p> <p>FMT_SMF.1 specifies the management functionality for each of the functional requirements in this Protection Profile. The management functionality is provided to the Administrator role.</p>
<p>O.MEDIATE</p> <p>The TOE must protect user data in accordance with its security policy.</p>	<p>FDP_ACC.2 FDP_ACF.1-NIAP-0407 FDP_IFC.2 FDP_IFF.1</p>	<p>FDP_ACC.2 specifies that the subjects and objects under control of the policy are to be defined and that all operations that involve access to (minimally) the data are controlled by the access control policy.</p> <p>FDP_ACF.1-NIAP-0407 details the manner in which objects are to be protected. The basics called for by the requirement are 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 ACI’s need to grant access in order for the subject to perform the operation on the object. The details of how the ACI’s are collected and the specific operations supported are specified in the Security Target, and with the attributes define the security policy to be enforced.</p> <p>FDP_IFC.2 requires that all information flow between subjects and objects be controlled by the information flow policy.</p> <p>FDP_IFF.2 specifies the protection required for the information flow control policy.</p>
<p>O.REPLAY_DETECTION</p> <p>The TOE will provide a means to detect and reject the replay of authentication data as well as other TSF data and security attributes.</p>	<p>FPT_RPL.1</p>	<p>FPT_RPL.1 meets this objective by requiring the TOE to detect and reject the attempted replay of authentication data, TSF data and security attributes.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.RESOURCE_SHARING</p> <p>The TOE shall provide mechanisms that mitigate attempts to exhaust the memory, computing and input/output resources provided by the TOE.</p>	<p>FMT_MOF.1 FMT_MTD.2 FRU_PRS.2 FRU_RSA.1</p>	<p>FMT_MOF.1 dictates the functionality required to manage the security functions of the TOE. One of these functions is the connection-oriented resource allocation parameters that prevent denial of service attacks. The ability to control this function is limited to the Administrator role.</p> <p>FMT_MTD.2 allows the Administrator to set the limits or quotas for controlled connection-oriented resources.</p> <p>FRU_PRS.2 requires subjects be assigned a priority and shareable resources be assigned based on priority. This will limit the impact of denial of service attacks on the most critical, highest priority operations.</p> <p>FRU_RSA.1 is used to mitigate potential resource exhaustion attempts.</p>
<p>O.ROBUST_ADMIN_GUIDANCE</p> <p>The TOE will provide administrators with the necessary information for secure delivery and management</p>	<p>ADO_DEL.2 ADO_IGS.1 AGD_ADM.1 AGD_USR.1 AVA_MSU.2</p>	<p>ADO_DEL.2 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 <i>clean</i> (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.</p> <p>The ADO_IGS.1 requirement ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. The</p> <p>AGD_ADM.1 requirement mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>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.</p> <p>The AGD_USR.1 is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since the non-administrative users of this TOE are limited to relying parties it is expected that the user guidance would discuss how the data validation authentication mechanism is used, and any instructions on authenticating to the TOE. The description of the use of these mechanisms would not have to be repeated in the administrator's guide.</p> <p>AVA_MSU.2 ensures that the guidance documentation is complete and can be followed unambiguously to ensure the TOE is not mis-configured in an insecure state due to confusing guidance.</p>
<p>O.ROBUST_TOE_ACCESS</p> <p>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.</p>	<p>FIA_AFL.1 FIA_ATD.1 FIA_SOS.1 FIA_UAU.2 FIA_UAU.4 FIA_UAU.7 FIA_UID.2 FTA_TSE.1 AVA_SOF.1</p>	<p>FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by all users. The requirement enables an Administrator settable threshold that triggers an alarm event when unauthorized users attempting to gain access to an authorized user's account by guessing authentication data. An Administrator can take action to identify and stop the unauthorized user once the alarm event occurs.</p> <p>FIA_ATD.1 defines the attributes of users including a unique userID and role membership. The role membership determines what access and actions an authenticated user is allowed to take. Additional attributes restrict access to certain time periods.</p> <p>FIA_SOS.1 requires two-factor authentication for Administrators and allows the Administrator to set the password complexity for users in the Operator and Display roles. Robust authentication</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>credential requirements increase the level of effort and skill required to gain unauthorized access to the TOE.</p> <p>FIA_UAU.2 requires all users to authenticate themselves to the TOE before any access or actions are considered by the TOE.</p> <p>FIA_UAU.4 requires the TOE to prevent reuse of authentication data related to an Administrator login. Each login must contain some unique data and the TOE must verify the authentication data has not been repeated. This prevents a threat agent from recording and reusing authentication data.</p> <p>FIA_UAU.7 specifies the TOE only provide an indication that the authentication succeeded or failed. Additional information, such as invalid userID, would help a threat agent potentially gain unauthorized access.</p> <p>FIA_UID.2 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any mediated function.</p> <p>FTA_TSE.1 requires access control for TOE session establishment based on user location, user role, time and date.</p> <p>The AVA_SOF.1 requirement is applied to the authentication mechanism. For this TOE, the strength of function specified is medium. This requirement ensures the developer has performed an analysis of the authentication mechanism to ensure the probability of guessing a user's authentication would require a high-attack potential.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.SELF_PROTECTION</p> <p>The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure.</p>	<p>FDP_SDI.2 FPT_FLS.1 FPT_ITI.1 FPT_RVM.1 FPT_PHP.2 FPT_SEP.2</p>	<p>FDP_SDI.2 will detect and alarm when object data is corrupted. A corruption in the object field device data, whether the corruption is malicious or accidental, could lead to incorrect action or inaction in the overall control system.</p> <p>FPT_FLS.1 preserves a secure state when failures of computing resources occur. This is a fail secure measure.</p> <p>FPT_ITI.1 identifies when imported TSF data is corrupted. This prevents corrupted security attributes from being used by the TSF.</p> <p>FPT_RVM.1 ensures 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-bypass requirement, the TSF could not be relied upon to completely enforce the security policies.</p> <p>FPT_PHP.2 identifies physical tampering that might compromise the TSF.</p> <p>FPT_SEP.2 ensures 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. FPT_SEP.2 was used to require that the cryptographic module be provided its own address space. This is necessary to reduce the impact of programming errors in the remaining portions of the TSF on the cryptographic module.</p>
<p>O.SOUND_DESIGN</p> <p>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.</p>	<p>ADV_FSP_(EXP).1 ADV_HLD_(EXP).1 ADV_INT_(EXP).1 ADV_LLD_(EXP).1 ADV_ARC_(EXP).1 ADV_RCR.1 ADV_SPM.1</p>	<p>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.</p> <p>ADV_INT_(EXP).1 ensures that the design of the TOE has been performed using good software engineering design principles that require a modular design of the TSF. Modular code increases the developer's understanding of the interactions within the</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>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_(EXP).2, AVA_VLA.3).</p> <p>ADV_SPM.1 requires the developer to provide an informal model of the security policies of the TOE. Modeling these policies helps understand and reduce the unintended side effects that occur during the TOE's operation that might adversely affect the TOE's ability to enforce its security policies.</p> <p>ADV_FSP_(EXP).1 requires that the interfaces to the TSF be completely specified. In this TOE, a complete specification of the network interface (including the network interface card) is critical in understanding what functionality is presented to untrusted users and how that functionality fits into the enforcement of security policies. Some network protocols have inherent flaws and users have the ability to provide the TOE with network packets crafted to take advantage of these flaws. The routines/functions that process the fields in the network protocols allowed (e.g., TCP, UPD, ICMP, directory-specific protocols such as LDAP) must fully specified: the acceptable parameters, the errors that can be generated, and what, if any, exceptions exist in the processing. The functional specification of the hardware interface (e.g., network interface card) is also extremely critical. Any processing that is externally visible performed by NIC must be specified in the functional specification. Having a complete understanding of what is available at the TSF interface allows one to analyze this functionality in the context of design flaws.</p> <p>ADV_HLD_(EXP).1 requires that a high-level design of the TOE be provided. This level of design describes the architecture of the TOE in terms of subsystems. It identifies which subsystems are responsible for making</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>and enforcing security relevant (e.g., anything relating to an SFR) decisions and provides a description, at a high level, of how those decisions are made and enforced. Having this level of description helps provide a general understanding of how the TOE works, without getting buried in details, and may allow the reader to discover flaws in the design. ADV_ARC_(EXP).1 addresses the non-bypass (FPT_RVM) and domain separation (FPT_SEP) aspects of the TSF, since these need to be analyzed differently from other functional requirements.</p> <p>The low-level design, as required by ADV_LLD_(EXP).1, provides the reader with the details of the TOE's design and describes at a module level how the design of the TOE addresses the SFRs. This level of description provides the detail of how modules interact within the TOE and if a flaw exists in the TOE's design, it is more likely to be found here rather than the high-level design. This requirement also mandates that the interfaces presented by modules be specified. Having knowledge of the parameters a module accepts, the errors that can be returned and a description of how the module works to support the security policies allows the design to be understood at its lowest level.</p> <p>ADV_RCR.1 is 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.</p>
<p>O.SOUND_IMPLEMENTATION</p> <p>The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.</p>	<p>ADV_IMP.2 ADV_INT_(EXP).1 ADV_LLD_(EXP).1 ADV_ARC_(EXP).1 ADV_RCR.1 ALC_TAT.1</p>	<p>While ADV_LLD_(EXP).1 (and ADV_ARC_(EXP).1 for the FPT_SEP and FPT_RVM aspects of the TSF) 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</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>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.</p> <p>While evaluators have the ability to “negotiate” the subset in ADV_IMP.1, ADV_IMP.2 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.</p> <p>When performing the activities associated with the ADV_INT_(EXP).1 requirement, the evaluators will ensure that the architecture of the implementation is modular and consistent with the architecture presented in the low-level design. Having a modular implementation provides the evaluators with the ability to more easily assess the accuracy of the implementation, with respect to the design. If the implementation is overly complex (e.g., circular dependencies, not well understood coupling, reliance on side-effects) the evaluator may not have the ability to assess the accuracy of the implementation.</p> <p>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</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>how the implementation representation is to be analyzed.</p> <p>ADV_RCR.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.</p>
<p>O.THOROUGH_FUNCTIONAL_TESTING</p> <p>The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.</p>	<p>ATE_COV.2 ATE_FUN.1 ATE_IND.2 ATE_DPT.2</p>	<p>In order to satisfy O.THOROUGH_FUNCTIONAL_TESTING, the ATE class of requirements is necessary. The component ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage. In addition, the developer must provide the test suite executables and source code, which are used for independently verifying the test suite results and in support of the test coverage analysis activities. ATE_COV.2 requires the developer to provide a test coverage analysis that demonstrates the TSFI are completely addressed by the developer's test suite. While exhaustive testing of the TSFI is not required, this component ensures that the security functionality of each TSFI is addressed. This component also requires an independent confirmation of the completeness of the test suite, which aids in ensuring that correct security relevant functionality of a TSFI is demonstrated through the testing effort. ATE_DPT.2 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. ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run</p>

Objectives	Requirements Addressing Objectives	Rationale
		<p>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.</p>
<p>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.</p>	<p>FMT_MTD.1 FPT_STM.1</p>	<p>FMT_MTD.1 helps satisfy this objective by providing the capability to set the time used for generating time stamps to an Administrator or a trusted IT entity, such as an NTP server.</p> <p>FPT_STM.1 requires that the TOE be able to provide reliable time stamps for its own use. 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.</p>
<p>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.</p>	<p>FTP_TRP.1</p>	<p>FTP_TRP.1 requires the TOE to provide a mechanism that creates a distinct communication path that protects the data that traverses this path from modification. This requirement ensures that the TOE can identify the communication end points and ensures that a user cannot insert themselves between the user and the TOE (man-in-the-middle) to modify or replace data without detection.</p>
<p>O.USER_GUIDANCE The TOE will provide users with the information necessary to correctly use the security mechanisms.</p>	<p>AGD_USR.1</p>	<p>The user guidance required by AGD_USR.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.</p>

Objectives	Requirements Addressing Objectives	Rationale
<p>O.VULNERABILITY_ANALYSIS_TEST</p> <p>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.</p>	<p>AVA_VLA.3</p>	<p>The AVA_VLA.3 component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated.</p> <p>AVA_VLA.3 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.</p>

640

6.3 Rationale for Assurance Requirements

The EAL definitions and assurance requirements in Part 3 of the Common Criteria and the CCEVS CIM for Medium Robustness^{MRBT} were reviewed and the *Medium Robustness Assurance Package* as defined in Section 5.2 was believed to best achieve the SOF goal.

645

6.4 Rationale for Strength of Function Claim

Part 1 of the Common Criteria defines “strength of function” in terms of the minimum efforts assumed necessary to defeat the expected security behavior of a TOE security function. There are three strength of function levels defined in Part 1: SOF-basic, SOF-medium and SOF-high. SOF-medium is the strength of function level chosen for this Protection Profile. The rationale for choosing SOF-medium was to be consistent with the Medium Robustness guidelines as described in Appendix D.

650

655 The TOE represents a high value target for critical infrastructure systems and is
implementing partial authorization which places it in the Medium Robustness portion of
the charts in Appendix D.

Appendix A: References

- 660 AGA AGA 12, Part 1, Cryptographic Protection of SCADA Communications
- CIDX CIDX Report on Cybersecurity Vulnerability Assessment Methodologies, version 2.0, November 2004
- 665 MRBT Consistency Instruction Manual for Development of US Government Protection Profiles for use in Medium Robustness Environments, NIAP Protection Profile Review Board, Release 3.0, 1 February 2005

Appendix B: Glossary

670	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 and the disclosure and modification of data.
675	Accountability	Property that allows activities in an IT system to be traced to the entity responsible for the activity.
	Administrator	A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.
680	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).
685	Asymmetric Key	The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.
690	Attack	An intentional act attempting to violate the security policy of an IT system.
	Authentication	Security measure that verifies a claimed identity.
695	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.
700	Availability	Timely, reliable access to IT resources.

	Compromise	Violation of a security policy.
	Confidentiality	A security policy pertaining to disclosure of data.
705	Critical Security Parameters	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.
710	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.
715	Cryptographic Boundary	An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.
720	Cryptographic Key	A parameter used in conjunction with a cryptographic algorithm that determines: <ul style="list-style-type: none">➤ 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.
725	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.
730	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 Protection Profile and additional rules imposed by the vendor.
735	DCS	A DCS is a type of plant automation system similar to a SCADA system, except that a DCS is usually employed in factories and is located within a more confined area. It uses a high-speed communications medium, which is usually a separate wire (network) from the plant LAN. ^{CIDX}

	Defense-in-Depth	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
740	Discretionary Access Control	A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. These controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
745	Embedded Cryptographic Module	One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).
	Enclave	A collection of entities under the control of a single authority and having a homogenous security policy. They may be logical, or may be based on physical location and proximity.
750	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.
755	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.
	IED	Any device incorporating one or more processors with the capability to receive or send data/control from or to an external source (e.g., electronic multifunction meters, digital relays, controllers). ^{AGA}
760	Integrity	A security policy pertaining to the corruption of data and TSF mechanisms.
	Integrity Label	A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.
765	Integrity Level	The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.
	Mandatory Access Control	A means of restricting access to objects based on subject and object sensitivity labels.
770	Mandatory Integrity Control	A means of restricting access to objects based on subject and object integrity labels.

Version 0.71 – May 18, 2006

	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.
775	Named Object	An object that exhibits all of the following characteristics: <ul style="list-style-type: none">➤ 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.
780		<ul style="list-style-type: none">➤ The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.
	Non-Repudiation	A security policy pertaining to providing one or more of the following:
785		<ul style="list-style-type: none">➤ 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.
790	Operating Environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
	Operational Key	Key intended for protection of operation information or for the production or secure electrical transmissions of key streams.
795	Peer TOE's	Mutually authenticated TOE's that interact to enforce a common security policy.
	PLC	A PLC is a hardened, special-purpose computer that was developed to replace relay-based control systems. ^{CIDX}
800	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

805		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.
810		High: Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.
	SCADA	A computer control system used in real time to monitor and control one or more remote facilities. The system collects data and/or sends control instructions, either automatically or by operators at other locations. SCADA is used to control facilities in industries such as telecommunications, water and waste control, energy, oil and gas refining, and transportation. ^{CIDX}
815		
	Secure State	Condition in which all TOE security policies are enforced.
	Security Attributes	TSP data associated with subjects, objects, and users that are used for enforcement of the TSP.
820		
	Security Level	The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.
	Sensitivity Label	A security attribute that represents the security level of an object and that describes the sensitivity (e.g. Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions.
825		
	Split Key	A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.
830		
	Subject	An entity within the TSC that causes operations to be performed.
	Symmetric Key	A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.
835	Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Version 0.71 – May 18, 2006

840	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.

845 **Appendix C: Acronyms**

	CCEVS	Common Criteria Evaluation and Validation Scheme
	CIM	Consistency Instruction Manual
	CM	Configuration Management
	DCS	Distributed Control System
850	EAL	Evaluation Assurance Level
	FIPS	Federal Information Processing Standard
	IED	Intelligent Electronic Device
	NIAP	National Information Assurance Partnership
	NIST	National Institute of Standards and Technology
855	NSA	National Security Agency
	PAC	Programmable Automation Controller
	PLC	Programmable Logic Controller
	RTU	Remote Terminal Unit
	SCADA	Supervisory Control and Data Acquisition
860	SFP	Security Function Policy
	SOF	Strength of Function
	TOE	Target of Evaluation
	TSC	TSF Scope of Control
	TSF	TOE Security Functions
865	TSP	TOE Security Policy

Appendix D: Robustness Environment Characterization

In trying to specify the environments in which TOE's 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.

870

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).

875

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.

880

Value of Resources

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor).

885

"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.

890

895

Authorization of Entities

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE area 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

900

905

910 allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database).

915 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 associate resources.

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

Selection of Appropriate Robustness Levels

930 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 selection relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

935 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.

940 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:

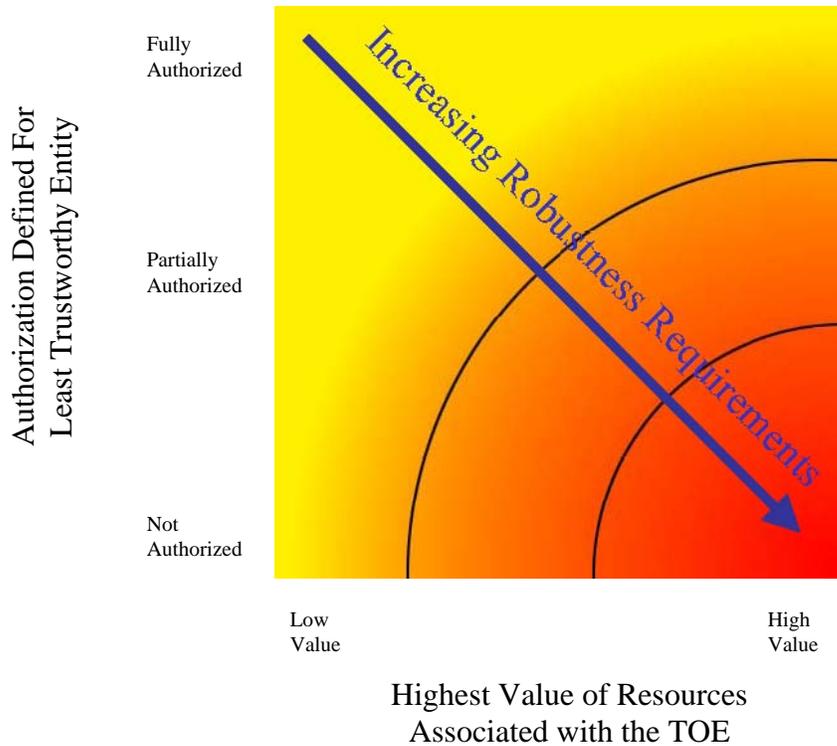
945 The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case,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

955 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.

965 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.

975 As depicted in the following figure, the robustness of the TOE’s 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.

985 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 not 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.



995

1000

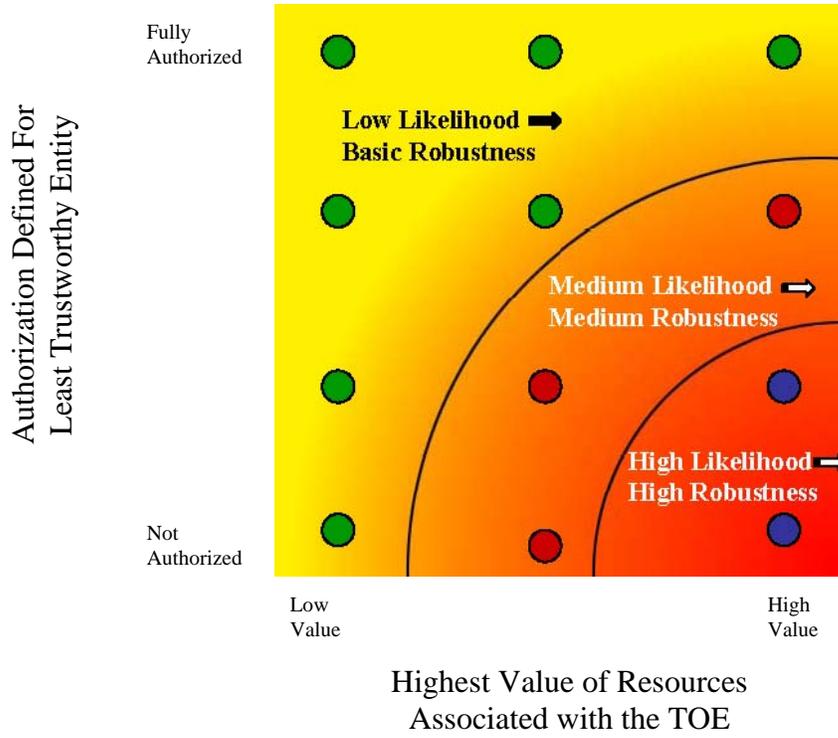
1005

1010

1015

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. Corresponding, 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 Protection Profile 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 below, corresponding to the likelihood that the 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 chose.

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 Protection Profile, the targeted threat level for a medium robustness TOE is characterized. This information is provided to help organizations using this Protection Profile ensure that the functional requirements specified by this medium robustness PP are appropriate for their intended application of a compliant TOE.



1020

1025

1030