GENERAL DYNAMICS
Mission Systems

Fortress Mesh Point ES210, ES520, ES820, ES2440 Security Target

Document Version

16-3723-R-0014

Version: 2.5

5/27/2016

Prepared By:

InfoGard Laboratories, Inc.
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401

Notices:

©2016 General Dynamics Mission Systems All rights reserved. All other brand names are trademarks, registered trademarks, or service marks of their respective companies or organizations.

It is prohibited to copy, reproduce or retransmit the information contained within this documentation without the express written permission of General Dynamics Mission Systems, 150 Rustcraft Road, Dedham, Massachusetts, 02026 USA.

Table of Contents

Tables

1. Security Target (ST) Introduction

- The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
- The ST reference shall uniquely identify the ST.
- The TOE reference shall identify the TOE.

The structure of this document is defined by CC v3.1r3 Part 1 Annex A.2, "Mandatory contents of an ST":

- Section 1 contains the ST Introduction, including the ST reference, Target of Evaluation (TOE) reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions

1.1 Security Target Reference

The Security Target reference shall uniquely identify the Security Target.

ST Title:	FORTRESS Mesh Point ES210, ES520, ES820, ES2440 Security Target
ST Version Number:	Version 2.5
ST Author(s):	Ryan Day
ST Publication Date:	5/27/2016
Keywords:	Network Device, IPsec, WLAN Access System

1.2 Target of Evaluation Reference

The Target of Evaluation reference shall identify the Target of Evaluation.

TOE Developer	General Dynamics Mission Systems	
	150 Rustcraft Road, Dedham, Massachusetts, 02026 USA	
TOE Name:	Fortress Mesh Point ES210, ES520, ES820, ES2440	
TOE HW Version:	ES210-3	810-00020-01
	ES210-4	810-00029-01

ES2440-0	810-00046-01
ES2440-34	810-00050-01
ES2440-3444	810-00038-01
ES2440-3444m	810-00060-01
ES2440-34m	810-00061-01
ES2440-35	810-00051-01
ES2440-3555	810-00037-01
ES520-34	810-00022-01
ES520-35	810-00015-01
ES820-34	810-00030-01
ES820-35	810-00023-01

TOE FW Version: 5.4.5.2240

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a Wireless Local Area Network (WLAN) Access Device. The TOE employs Mesh networking, which allows multiple TOEs to network within the operational environment. Only WLAN functionality is evaluated in this Security Target. All VPN Gateway functionality was evaluated in a separate Security Target.

1.3.2 TOE Usage

The TOE brings secure wireless communications to environmentally challenging situations, including, outdoor locations, and across long distances through a self-forming, self-healing mesh network. Delivered in a form factor that is rugged, weatherized, and easy to set-up and operate the TOE functions as both a wireless access point and bridge, with up to four powerful radios for maximum range and performance. The TOE has the following services available for usage: SSH, HTTPS, Console Port, SNMP, IPSec, Wireless Clients (WPA/WPA2), 802.1x, DNS, RADIUS, NTP (client only), Port 4949 (Mesh Viewer Protocol (MVP)). NOTE: This is not a list of evaluated services, only the services specifically discussed in Sections 1.4.2, and 6 have been evaluated.

1.3.3 TOE IT environment hardware/software/firmware requirements

- Hardware/Firmware Requirements
  - RS-232 Console Port compatible with the following enumeration settings:
    - bits per second: 9600
    - data bits: 8
    - parity: none
    - stop bits: 1
    - hardware flow control: none
  - Ethernet Client Hardware Requirements:
    - 10BASE-T/100BASE-TX Base Ethernet
  - Wireless Client Hardware/Firmware Requirements:

        - Wireless 2.4GHz, 4.4GHz, 4.9GHz, or 5.0GHz, IEEE 802.11 a/b/g/n (depending on radio see Section for Radio Configuration)
        - WPA2 (a security protocol and security certification program developed by the Wi-Fi Alliance to secure wireless computer networks)
    - Antenna:
        - ES210 and ES2440 Specific (not in ES520, 820):
            - GPS antenna with SMA connector
        - Wi-Fi Antenna with N-style connector
        - Capable of transmitting and receiving on the required frequency as described by the Section for Radio Configuration.
- Software Requirements:
    - Syslog server
        - Compatible with RFC 3164
        - Supporting IPsec as defined in FCS_IPSEC_EXT.1 IPsec
    - RADIUS server
        - Compatible with RFC 2865
        - Supporting IPsec as defined in FCS_IPSEC_EXT.1 IPsec
    - NTP server
        - V4 conformant to RFC 5905 with a SHA-1 authentication[1]
    - GUI access
        - Firefox v3.6 to 44.0.2
        - IE version 7.0-10.0
        - Compatible with HTTPS implementing:
            - HTTPS protocol that complies with RFC 2818
            - TLS 1.0 (RFC 2246)
        - Compatible with TLS using the following:
            - Mandatory cipher suites:
                - TLS_RSA_WITH_AES_128_CBC_SHA
            - Optional cipher suites:
                - TLS_RSA_WITH_AES_256_CBC_SHA
                - TLS_DHE_RSA_WITH_AES_128_CBC_SHA
                - TLS_DHE_RSA_WITH_AES_256_CBC_SHA
    - SSH
        - V2 client compatible with the list of required ciphers (as listed in Section FCS_SSH_EXT.1 SSH)

1.4 Target of Evaluation Description

1.4.1 Target of Evaluation Physical Boundaries

The TOE, Fortress Mesh Point, is a VPN gateway device that provides secure wireless communications for their intended environment.

Table 1 – TOE Processor Identification

Model	Processor	Crypto Accelerator

[1] SHA-1 authentication for NTP was not evaluated and therefore cannot claim any cryptographic security.

ES210	AMD Alchemy AU1550	Xilinx Spartan FPGA
ES820	AMD Alchemy AU1550	Xilinx Spartan FPGA
ES520	AMD Alchemy AU1550	Xilinx Spartan FPGA
ES2440	Broadcom XLS416	Xilinx Spartan FPGA

The following table summarizes the use of Ethernet ports at the physical boundary of the TOE for the different models.

Table 2 – TOE Ethernet Port Summary

Model	# of Eth Ports	HW Label	GUI Label	Takes PoE	Serves PoE
ES210	2	Ethernet (WAN)	Ethernet1	no	no
		Ethernet	Ethernet2	no	no
ES820	2	Enet1/P1	Ethernet1	no	no
		Enet2/P2	Ethernet2	no	no
ES520	9	WAN	wan1	yes	no
		1–8	lan1–lan8	no	yes
ES2440	3	Ethernet1/WAN/POE	Ethernet1	yes	no
		Ethernet2	Ethernet2	no	no
		Ethernet3	Ethernet3	no	no

1.4.1.1 Radio Configurations

The TOE radio modules are logically identical and have no implications on security or functionality except the frequency and the link layer (layer 1 on the OSI stack) which are specific to the radio. Within each unique identifier there is a primary model number (ES2440) followed by a dash and then a digit (3, 4, or 5).

- Radio '3' - 250mW frequencies 2.4GHz, 4.9GHz and 5GHz using 802.11a/b/g/n
- Radio '4' - 600mW frequency 4.4GHz and 802.11 a/n
- Radio '5' -- 500mW frequencies 4.9GHz, 5GHz using 802.11 a/n

The guidance documentation that is part of the TOE is listed in Section 10, "References," within Table 15: TOE Guidance Documentation.

1.4.1.2 Physical Boundary Description

1.4.1.2.1 ES210

The ES210 acts as a 2-layer bridge with VPN functionality and a wireless access point. The ES210 can operate at the given frequencies and data link protocols listed above in section 1.4.1.1 Radio Configuration. The physical boundaries of the ES210 are at all of the connectors of the TOE module:

- RJ45 10/100BT Ethernet Port (2)
  - Provides a port for the user to access the network as well as allows access to the management functionality with administrative user authentication. The only difference between the two ports is that the port labeled (Ethernet1/WAN) is encrypted by default, the other is not.
- 3 Pin Con-X Serial Connector (3 pin mil-spec round connector)

    - Local CLI management interface
- 2 Pin Con-X Power Connector (2 pin mil-spec round connector)
    - Provides power to the ES210
- RP-TNC Antenna Connector (1)
    - For the various antenna options described in Section Radio Configuration.
- SMA Connector
    - GPS antenna

Indicators are used to allow the operator to have a quick indication of the state of the ES210:

- Power
    - Indicates the power status of the TOE
- Battery
    - Indicates the charge state of the battery
- Ethernet1/Ethernet 2 – Link/Activity
    - Indicates the status and activity of the Ethernet port
- Radio activity
    - Indicates activity on that radio position

The ES210 also has the following physical button controls:

- Power On/Off
    - Allows the device to be powered
- Blackout Mode
    - Turns off all LED indicators
- RF Kill
    - Turns all radio transmissions off
- Zeroize
    - Restores factory defaults

1.4.1.2.2 ES520

The ES520 acts as a 2-layer bridge with VPN functionality and a wireless access point. The ES520 can operate at the given frequencies and data link protocols listed above in section 1.4.1.1 Radio Configuration. The physical boundaries of the ES520 are at all of the connectors of the TOE module:

- RJ45 10/100BT Ethernet Port (8)
    - Provides a port for the user to access the network as well as allows access to the management functionality with administrative user authentication. The only difference between the two ports is that the port labeled (WAN) is encrypted by default, the other is not.
- USB Host Connector
    - This is excluded in the CC evaluated configuration
- 10/100BT WAN Port (1)
    - Provides a port for the user to access the network as well as allows access to the management function with administrative user authentication
- 3 Pin Con-X Serial Connector (3 pin mil-spec round connector)
    - Local CLI management interface
- DC Power Input Connector
    - Provides power to the ES520
- N-type Antenna Connector (2)

  - ES520
  - For the various antenna options described in Section Radio Configurations

Indicators are used to allow the operator to have a quick indication of the state of the ES520:

- Power
  - Indicates the power status of the TOE
- Clr
  - Excluded
- Status 1
  - Indicates system status
- Status 2
  - Excluded
- Fail
  - Excluded
- Radio1/Radio2 (Upper)
  - Indicates the activity on the radio
- Radio1/Radio2 (Lower)
  - Excluded

The ES520 also has the following controls:

- Reset Button
  - Power cycles the TOE

1.4.1.2.3 ES820

The ES820 acts as a 2-layer bridge with VPN functionality and a wireless access point. The ES820 can operate at the given frequencies and data link protocols listed above in section 1.4.1.1 Radio Configuration. The physical boundaries of the ES820 are at all of the connectors of the TOE module:

- MIL Connector; includes the following interfaces:
  - RJ45 10/100BT Ethernet Port (2)
    - Provides a port for the user to access the network as well as allows access to the management functionality with administrative user authentication. The only difference between the two ports is that the port labeled (WAN) is encrypted by default, the other is not.
  - USB
    - This is excluded in the CC evaluated configuration
  - Serial
    - Local CLI management interface
  - All LED indicators
  - All Controls
- 3 Pin Con-X Serial Connector (3 pin mil-spec round connector)
  - Supplies power to the TOE
- N-type Antenna Connector (2)
  - ES820
  - For the various antenna options described in Section Radio Configurations

Indicators are used to allow the operator to have a quick indication of the charge state of the ES820. The following indicators are available through the MIL connector:

- Power

  - Indicates the power status of the TOE
- Status
  - Excluded
- Ethernet1/Ethernet 2 – Link/Activity
  - Indicates the status and activity of the Ethernet port
- Radio activity
  - Indicates activity on that radio position

The ES820 has the following input functions by means of the MIL connector:

- Power On/Off
  - Allows the device to be powered
- Blackout Mode
  - Turns off all LED indicators
- RF Kill
  - Turns all radio transmissions off
- Reset
  - Power cycles the device
- Zeroize
  - Restores factory defaults

1.4.1.2.4 ES2440

The ES2440 acts as a 2-layer bridge with VPN functionality and a wireless access point. The ES2440 can operate at the given frequencies and data link protocols listed above in section 1.4.1.1 Radio Configuration. The physical boundaries of the ES2440 are at all of the connectors of the TOE module:

- RJ45 10/100/1000BT Ethernet Port (3)
  - Provides a port for the user to access the network as well as allows access to the management functionality with administrative user authentication. The only difference between the first port and the other two ports is that the port labeled (Ethernet1/WAN/POE) allows power over Ethernet (802.3af), and the others do not.
- RJ45 Serial Connector
  - Local CLI management interface
- 2 Pin Con-X Power Connector (2 pin mil-spec round connector)
  - Provides power to the ES2440
- N-type Antenna Connector (8)
  - For the various antenna options described in Section Radio Configurations
- SMA Connector
  - GPS antenna

Indicators are used to allow the operator to have a quick indication of the state of the ES2440:

- Power
  - Indicates the power status of the TOE
- Ethernet1/Ethernet 2/Ethernet3 link/activity – Link/Activity
  - Indicates the status and activity of the Ethernet port
- Radio1/Radio2/Radio3/Radio4 activity
  - Indicates activity on that radio position

The ES2440 also has the following physical button controls:

- Recessed Button
    - Restores factory defaults

1.4.2 Target of Evaluation Description and Logical Boundaries

The logical boundary of the TOE includes those security functions implemented exclusively by the TOE running on the software version SW: 5.4.5.2240. The commands listed in Section 1.24 of the Operational Guidance [11] cover the relevant functionality required to meet this ST. Section 1.25 of the Operational Guidance [11] discuss prohibited and compliant functionality.

1.4.2.1 Audit

The TOE has the ability to audit events based on a specified criteria. To protect the TSF from audit log overflow, the TOE uploads audit data to an external syslog server through an IPsec tunnel. The audit record includes: the date and time of the event, the user who triggered the event (if event was user based and user is known), and event specific information. A subset of auditable events required by this ST is found in FAU_GEN and Table 12 – Audit Record Events. The TOE also protects all locally stored audit data from un-authorized modification and deletion. The TOE implements SyslogD version 1.5.0.

1.4.2.2 Cryptographic Operations

The TOE provides cryptographic functions to protect information, including mechanisms to encrypt, decrypt, hash, digitally sign, and perform cryptographic key agreement. The evaluated configuration uses a subset of the cryptographic implementations listed in Section 9 for all cryptographic purposes. The FIPS-Approved cryptographic algorithms used by the TOE, and specified by the SFRs, are listed in Table 15. The following protocols are implemented by the TOE and use FIPS-Approved cryptographic algorithms:

- WPA2 (802.11i)
- WPA2 (EAP-TLS)
- IPsecTLS1.0/HTTPS
- SSHv2
- HTTPS/TLS

1.4.2.3 User Data Protection

The TOE protects user data, (i.e., only that data exchanged with wireless client devices), using the IEEE 801.11i standard wireless security protocol. The TOE mediates the flow of information passing to and from the WAN port and ensures that resources used to pass network packets through the TOE do not contain any residual information.

1.4.2.4 Identification and Authentication

The TOE requires the system administrators be authenticated before access to the TOE is granted; administrators may login to the TOE by providing a user name and password via a local RJ45 using a serial RS-232 connection, and via SSH, HTTPS, or X.509 for TLS. Administrators may connect to the TOE remotely via the LAN, WAN, or 802.11a/b/g/n interfaces.

The TOE displays a configurable access banner and requires an administrator to authenticate using a username and password. An external RADIUS server can be configured for authentication through an IPsec tunnel. Authentication can take place, by user name and password (and hexadecimal device ID if

applicable). For IPsec, the TOE also supports X.509 certificates. EAP-TLS is used for WPA2 wireless authentication via x.509 certificates.

1.4.2.5 Security Management

The management of the security relevant parameters of the TOE must be performed by the authorized administrator; the TOE provides the following management interfaces:

- Command Line Interface (CLI) via
  - local RJ45 or serial connection,
  - Remote SSH interface via the LAN, WAN ports, and 802.11 wireless interface
- Remote HTTPS Web UI via the LAN, WAN ports, and 802.11 wireless interface

1.4.2.6 Protection of the TSF

The TOE identification and authentication security functions allow only authenticated administrative users direct access to the TOE. If a wireless user does not authenticate as an administrative user then that user is a wireless client and can only pass traffic through the TOE and cannot execute commands on the TOE.

Administrative users are allowed to login via the CLI and Web UI to access all management functions. The management interfaces do not allow administrative users access to the underlying operating system and there are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE. Any access to a management interface (CLI or GUI) is protected by a secure channel except via RS-232; as this is considered local administration.

The TOE has the capability to obtain reliable time from a remote Network Time Protocol (NTP) Server to provide reliable time stamps for audit services. Additionally, the system administrator can manually set the time (maintained locally in the hardware Real Time Clock (RTC)) on the TOE using the Web UI or CLI management interfaces.

The TOE runs a set of self-tests on power-on to verify the correct operation of the TOE's underlying hardware, TOE software and cryptographic modules. Additional cryptographic tests are performed during normal operation. The security of network data is maintained by ensuring no residual information is included in network packets.

1.4.2.7 TOE Access

The TOE displays the access banner before establishing an administrative session. The TOE terminates an interactive session after an Authorized Administrator-configurable time interval of session inactivity. A wireless client session is defined as being allowed access to a particular port on the application layer. The TOE is able to deny establishment of a wireless client session based mac address.

1.4.2.8 Trusted Path/Channels

The TOE uses 802.11-2007 and IPsec to provide a trusted communication channel between itself and any authorized IT entities. In addition to IPsec, EAP-TLS is used for RADIUS.

The TSF initiates communication via the trusted channel for RADIUS, NTP and Syslog. The TOE uses SSH and TLS/HTTPS to provide a trusted communication path between itself and remote administrators.

1.4.2.9 Excluded Functionality

The TOE includes the following functionality that may not be enabled or used in in the CC evaluated configuration:

- SNMP

1.4.2.10 Unevaluated Features

The TOE includes the following functionality that is not covered this Security Target and the associated evaluation:

- VPN Gateway functionality (evaluated in a separate evaluation)
- GPS
- DHCP server
- DNS services
- QoS
- VLANs
- Mobile Security Protocol (MSP)
- Device Access Control
- Fortress Mesh Viewer Protocol
- Layer 2 link management (e.g. Spanning Tree Protocol)

These features may be used in the evaluated configuration; however, no assurance as to the correct operation of these features is provided.

1.5 Notation, formatting, and conventions

The notation, formatting, and conventions used in this security target are defined below; these styles and clarifying information conventions were developed to aid the reader.

Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification. Those notes specific to the TOE are marked “TOE Application Note;” those taken from the Protection Profile are marked “PP Application Note;”.

The notation conventions that refer to iterations, assignments, selections, and refinements made in this security target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a protection profile.

The notation used in those PP to indicate iterations, assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document. Additionally, obvious errors in the PP are corrected and noted as such.

The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; paragraph 6.4.1.3.2, “Permitted operations on components” as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

Iterations are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1).

Assignments made by the ST author are identified with **bold text.**

Selections are identified with underlined text**.** Selections within selections are identified with double underlined text.

Refinements that add text use ***bold and italicized text*** to identified the added text***.*** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

2. Conformance Claims

2.1 Common Criteria Conformance Claims

This Security Target is conformant to the Common Criteria Version 3.1r3, CC Part 2 extended [10], and CC Part 3 extended [11].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the Wireless Local Area Network (WLAN) Access Systems Protection Profile, Version 1.0, dated December 1, 2011 [14]. This Protection Profile will be referred to as WLANAS or PP for convenience throughout this Security Target.

2.3 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements package, neither as package-conformant or package-augmented.

2.4 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
    - All threats defined in the PP are carried forward to this ST;
    - No additional threats have been defined in this ST.
- Organizational Security Policies
    - All OSP defined in the PP are carried forward to this ST;
    - No additional OSPs have been defined in this ST.
- Assumptions
    - All assumptions defined in the PP are carried forward to this ST;
    - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
    - All objectives defined in the PP are carried forward to this ST.
- All SFRs and SARs defined in the PP are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the PP have been properly instantiated in this Security Target; therefore, this ST shows exact compliance to the PP.

3. Security Problem Definition

3.1 Threats

The following table defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP unchanged.

Table 3: Threats

Threat	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. 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. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.
T.RESOURCE_EXHAUSTION	A process or user may deny access to TOE services by exhausting critical resources on the TOE.

3.2 Organizational Security Policies

The following table defines the organizational security policies which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP unchanged.

Table 4: Organizational Security Policies

OSP	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.COMPATIBILITY	The TOE must meet Request for Comments (RFC) requirements for implemented protocols to facilitate inter-operation with other network equipment (e.g., certificate authority, NTP server) using the same protocols.
P.EXTERNAL_SERVERS	The TOE must support standardized (RFCs) protocols for communication with a centralized audit server and a RADIUS authentication server.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ADMIN_ACCESS	Administrators shall be able to administer the TOE both locally and remotely through protected communications channels.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP unchanged.

Table 5: Assumptions

Assumption	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
A.NO_TOE_BYPASS	Information cannot flow between the wireless client and the internal wired network without passing through the TOE.

4. Security Objectives

4.1 Security Objectives for the TOE

Table 6: Security Objectives for the TOE

TOE Objective	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
O.CRYPTOGRAPHIC_FUNCTIONS	The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.
O.FAIL_SECURE	The TOE shall fail in a secure manner following failure of the power-on self-tests.
O.AUTH_COMM	The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity.
O.PROTOCOLS	The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability, that also support communication with a centralized audit server and a RADIUS authentication server.
O.REPLAY_DETECTION	The TOE will provide a means to detect and reject the replay of authentication data and other TSF data and security attributes.
O.RESOURCE_AVAILABILITY	The TOE shall provide mechanisms that mitigate user

Table 6: Security Objectives for the TOE	
TOE Objective	Description
	attempts to exhaust TOE resources (e.g., persistent storage).
O.ROBUST_TOE_ACCESS	The TOE will provide mechanisms that control an administrator's logical access to the TOE and to control administrative access from a wireless client.
O.TIME_STAMPS	The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these timestamps.
O.WIRELESS_CLIENT_ACCESS	The TOE will provide the capability to restrict a wireless client in connecting to the TOE.

4.2 Security Objectives for the Operational Environment

Table 7: Security Objectives for the Operational Environment	
Objective	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.NO_TOE_BYPASS	Information cannot flow between external and internal networks located in different enclaves without passing through the TOE.

5. Extended Components Definition

This section provides definition of the extended security functional and assurance requirements; the components that are CC Part 2 extended, and CC Part 3 extended, i.e., NIAP interpreted requirements, and extended requirements.

5.1 Extended Security Functional Requirements Definitions

There are no extended Security Functional Requirements defined in this Security Target. All extended SFRs were taken from the PP.

5.2 Extended Security Assurance Requirement Definitions

There are no extended Security Assurance Requirements defined in this Security Target. All extended SARs were taken from the PP.

6. Security Requirements

This section describes the security functional and assurance requirements for the TOE; those that are CC Part 2 conformant, CC Part 2 extended, CC Part 3 conformant, and CC Part 3 extended.

6.1 Security Function Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the PP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 8: Security Functional Requirements

#	SFR	Description
1	FAU_GEN.1	Audit Data Generation
2	FAU_GEN.2	User Audit Association
3	FAU_SEL.1	Selective Audit
4	FAU_STG.1	Protected Audit Trail Storage (Local Storage)
5	FAU_STG_EXT.1	External Audit Trail Storage
6	FAU_STG_EXT.3	Action in Case of Loss of Audit Server Connectivity
7	FAU_SAR.1	Audit Review
8	FAU_SAR.2	Restricted Audit Review
9	FAU_STG_EXT.4	Prevention of Audit Data Loss
10	FCS_CKM.1(1)	Cryptographic Key Generation (Symmetric Keys for WPA2 Connections)
11	FCS_CKM.1(2)	Cryptographic Key Generation (Asymmetric Keys)
12	FCS_CKM.2(1)	Cryptographic Key Distribution (PMK)
13	FCS_CKM.2(2)	Cryptographic Key Distribution (GTK)
14	FCS_CKM_EXT.4	Cryptographic Key Zeroization
15	FCS_COP.1(1)	Cryptographic Operation (Data Encryption/Decryption)
16	FCS_COP.1(2)	Cryptographic Operation (Cryptographic Signature)
17	FCS_COP.1(3)	Cryptographic Operation (Cryptographic Hashing)
18	FCS_COP.1(4)	Cryptographic Operation (Keyed-Hash Message Authentication)
19	FCS_COP.1(5)	Cryptographic Operation (WPA2 Data Encryption/Decryption)
20	FCS_IPSEC_EXT.1	Extended: Internet Protocol Security (IPsec) Communications
21	FCS_TLS_EXT.1	Transport Layer Security
22	FCS_SSH_EXT.1	Secure Shell
23	FCS_HTTPS_EXT.1	HTTP Security
24	FCS_RBG_EXT.1	Extended: Cryptographic Operation: Random Bit Generation
25	FDP_RIP.2	Full Resident Information Protection
26	FIA_AFL.1	Authentication Failure Handling

Table 8: Security Functional Requirements	
# SFR	Description
27 FIA_PMG_EXT.1	Password Management
28 FIA_PSK_EXT.1	Extended: Pre-Shared Key Composition
29 FIA_UIA_EXT.1	User Identification and Authentication
30 FIA_UAU_EXT.5	Extended: Password-based Authentication Mechanisms
31 FIA_UAU.6	Re-authenticating
32 FIA_UAU.7	Protected Authentication Feedback
33 FIA_X509_EXT.1	Extended: X.509 Certificates
34 FIA_8021X_EXT.1	Extended: 802.1X Port Access Entity (Authenticator) Authentication
35 FMT_MOF.1	Management of Security Functions Behavior
36 FMT_MTD.1(1)	Management of TSF Data (General TSF Data)
37 FMT_MTD.1(2)	Management of TSF Data (Reading of Authentication Data)
38 FMT_MTD.1(3)	Management of TSF Data (for reading of all symmetric keys)
39 FMT_SMF.1	Specification of management functions
40 FMT_SMR.1	Security Management Roles
41 FPT_FLS.1	Fail Secure
42 FPT_RPL.1	Replay Detection
43 FPT_STM.1	Reliable Time Stamp
44 FPT_TUD_EXT.1	Extended: Trusted Update
45 FPT_TST_EXT.1	Extended: TSF Testing
46 FRU_RSA.1	Maximum Quotas
47 FTA_SSL_EXT.1	TSF-initiated session locking
48 FTA_SSL.3	TSF-initiated termination
49 FTA_SSL.4	User-initiated termination
50 FTA_TAB.1	Default TOE Access Banners
51 FTA_TSE.1	TOE Session Establishment
52 FTP_ITC.1	Inter-TSF trusted channel
53 FTP_TRP.1	Trusted Path

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

The TSF shall be able to generate an audit record for the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions;
d) Specifically defined auditable events listed in Table 12.

PP Application Note:

The ST author can include other auditable events directly in the table; they are not limited to the list presented.

Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is present in Table 12 – Audit Record Events.

Assurance Activity:

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 12 – Audit Record Events.

The evaluator shall in particular ensure that the operational guidance is clear in relation to the contents for failed cryptographic events. In Table 12 – Audit Record Events, information detailing the cryptographic mode of operation and a name or identifier for the object being encrypted is required. The evaluator shall ensure that name or identifier is sufficient to allow an administrator reviewing the audit log to determine the context of the cryptographic operation (for example, performed during a key negotiation exchange, performed when encrypting data for transit) as well as the non-TOE endpoint of the connection for cryptographic failures relating to communications with other IT systems.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The TOE may contain functionality that is not evaluated in the context of this PP because the functionality is not specified in an SFR. This functionality may have administrative aspects that are described in the operational guidance. Since such administrative actions will not be performed in an evaluated configuration of the TOE, the evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP, which thus form the set of “all administrative actions”. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

The evaluator shall test the TOE’s ability to correctly generate audit records by having the TOE generate audit records in accordance with the assurance activities associated with the functional requirements in this PP. Additionally, the evaluator shall test that each administrative action applicable in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing to ensure the TOE can detect replay attempts will more than likely be done to demonstrate that requirement FPT_RPL.1 is satisfied. Another example is that testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

FAU_GEN.1.2

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 9.

PP Application Note:

As with the previous component, the ST author should update Table 9 above with any additional information generated. "Subject identity" in the context of this requirement could either be the administrator's user id or the affected network interface, for example.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

6.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1

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.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

6.1.1.3 FAU_SEL.1 Selective Audit

FAU_SEL.1.1 [2]

The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

~~a) administrator identity;~~ [3]
b) event type;
c) success of auditable security events;
d) failure of auditable security events; and
e) User Interface
f) Encrypted/Clear Zone

PP Application Note:
The intent of this requirement is to identify all criteria that can be selected to trigger an audit event. For the ST author, the assignment is used to list any additional criteria or "none". The auditable event types are listed in Table 11: Audit Record Events.

Assurance Activities:

[2] Logs are filtered only when being sent to an external audit log server, all logs are stored locally regardless of selection settings. In addition, the TOE itself cannot filter C and D above, however this information is in the body of the logs, and the user is easily able to do this with a unix-style "grep" statement.

[3] Per TD0010, Administrator Identity has been removed.

The evaluator shall review the administrative guidance to ensure that the guidance itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The administrative guidance shall also contain instructions on how to set the pre-selection, as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced.

The evaluator shall also perform the following tests:

a) Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.

b) Test 2 [conditional]: If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

6.1.1.4 FAU_STG.1 Protected Audit Trail Storage (Local Storage)

FAU_STG.1.1

The TSF shall protect **3.5 MBytes** locally stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2

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

PP Application Note:
In addition to the capability to export the audit information, the TOE is required to have some amount of local storage. The ST writer completes the assignment with the amount of local storage available for the audit records; this can be in megabytes, average number of audit records, etc.

Assurance Activity:

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

6.1.1.5 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1.1

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel implementing the IPsec protocol.

PP Application Note:

The TOE also relies on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records, the storage of these audit records and the ability to allow the administrator to

review these audit records is provided by the operational environment. The ST author chooses the means by which this connection is protected using the selection. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.

Assurance Activity:

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

- Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

6.1.1.6 FAU_STG_EXT.3 Action in Case of Loss of Audit Server Connectivity

FAU_STG_EXT.3.1

The TSF shall **stop sending packets to the syslog server, and add a "Communication error" message to the local log** if the link to the external IT entity collecting the audit data generated by the TOE is not available.

PP Application Note:

The ST author fills in the action the TOE takes (e.g. pages the administrator, stops passing packets) if a link to the audit server is unavailable.

Assurance Activity:

The evaluator shall examine the administrative guidance to ensure it instructs the administrator how to establish communication with the audit server. The guidance must instruct how this channel is established in a secure manner (e.g., IPsec, TLS). The evaluator checks the administrative guidance to determine what action(s) is taken if the link between the TOE and audit server is broken. This could be due to network connectivity being lost, or the secure protocol link being terminated.

The evaluator shall examine the operational guidance to determine any activities that must take place after connectivity is restored to ensure that local audit events captured during the period of loss are synchronized with the audit trail on the audit server, and informs the administrator of any limitations on the data that are able to be sent (for instance, if the duration of the outage is significant, the local store may not contain all of the records that where generated during this period).

The evaluator shall perform the following test for this requirement:

- Test 1: The evaluator shall test the administrative guidance by establishing a link to the audit server. Note that this will need to be done in order to perform the assurance

activities prescribed under FAU_GEN.1. The evaluator shall disrupt the communication link (e.g., unplug the network cable, terminate the protocol link, shutdown the audit server) to determine that the action(s) described in the administrative guide appropriately take place.

6.1.1.7 FAU_SAR.1 Audit Review

FAU_SAR.1.1

The TSF shall provide Authorized Administrators with the capability to read all audit data from the audit records.

FAU_SAR.1.2

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

6.1.1.8 FAU_SAR.2 Restricted Audit Review

FAU_SAR.2.1

The TSF shall prohibit all users read access to the audit records in the audit trail, except Authorized Administrators.

6.1.1.9 FAU_STG_EXT.4 Prevention of Audit Data Loss

FAU_STG_EXT.4.1

The TSF shall provide the Authorized Administrator the capability to select one or more of the following actions:

- prevent auditable events, except those taken by the Authorized Administrator, and
- overwrite the oldest stored audit records

to be taken if the audit trail is full.

PP Application Note:

The TOE provides the Authorized Administrator the option of preventing audit data loss by preventing auditable events from occurring. The Authorized Administrator actions under these circumstances are not required to be audited. The TOE also provides the Authorized Administrator the option of overwriting "old" audit records rather than preventing auditable events, which may protect against a denial-of-service attack.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1(1) Cryptographic Key Generation (Symmetric Keys for WPA2 Connections)

FCS_CKM.1.1(1)

The TSF shall derive symmetric cryptographic keys in accordance with a specified cryptographic key derivation algorithm PRF-384 with specified cryptographic key size 128 bits using a Random Bit Generator as specified in FCS_RBG_EXT.1 and that meet the following: 802.11-2007.

PP Application Note:

This requirement applies only to the keys that are generated/derived for the communications between the access point and the client once the client has been authenticated. It refers to the generation of the GTK (through the RBG specified in this PP) as well as the derivation of the PTK from the PMK, which is done using a random value generated by the RBG specified in this PP, the HMAC function using SHA-1 as specified in this PP, as well as other information. This is specified in 802.11-2007 primarily in chapter 8.

Assurance Activity:

The cryptographic primitives will be verified through assurance activities specified later in this PP. The evaluator shall verify that the TSS describes how the primitives defined and implemented by this PP are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. The TSS shall also provide a description of the developer's method(s) of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also any third-party testing that is performed. The evaluator shall ensure that the description of the testing methodology is of sufficient detail to determine the extent to which the details of the protocol specifics are tested

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

FCS_CKM.1.1(2)

The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with a:

- NIST Special Publication 800-56A, "recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;
- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, "Digital Signature Standard")
- NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

PP Application Note:

This component requires that the TOE be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., IPsec). If multiple schemes are supported, then the ST author should iterate this requirement to capture this capability. The scheme used will be chosen by the ST author from the selection.

Since the domain parameters to be used are specified by the requirements of the protocol in this PP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies to the protocols specified in this PP.

The generated key strength of 2048-bit DSA and rDSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

Assurance Activity:

The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.
In order to show that the TSF implements complies with 800-56A and/or 800-56B, depending on the selections made, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.
- For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to "shall" or “should” statements shall be described;
- Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.

6.1.2.3 FCS_CKM.2(1) Cryptographic Key Distribution (PMK)

FCS_CKM.2.1(1)

The TSF shall distribute the 802.11 Pairwise Master Key in accordance with a specified cryptographic key distribution method: receive from 802.1X Authorization Server that meets the following: 802.11-2007 and does not expose the cryptographic keys.

PP Application Note:

This requirement applies to the Pairwise Master Key that is received from the RADIUS server by the TOE. The intent of this requirement is to ensure conformant TOEs implement 802.1X authentication prior to establishing secure communications with the client in addition to disallowing implementations that only support pre-shared keys. Because communications with the RADIUS server are required to be performed over an IPsec-protected connection, the transfer of the PMK will be protected.

Assurance Activity:

The evaluator shall examine the TSS to determine that it describes how the PMK is transferred (that is, through what EAP attribute) to the TSF.

The evaluator shall perform the following test:

- Test 1: The evaluator shall establish a session between the TOE and a RADIUS server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the RADIUS server and the TOE during a successful attempt to connect a wireless client to the TOE to determine that the PMK is not exposed.

6.1.2.4 FCS_CKM.2(2) Cryptographic Key Distribution (GTK)

FCS_CKM.2.1(2)

The TSF shall distribute Group Temporal Key in accordance with a specified cryptographic key distribution method: AES Key Wrap in an EAPOL-Key frame that meets the following: RFC 3394 for AES Key Wrap, 802.11-2007 for the packet format and timing considerations and does not expose the cryptographic keys.

PP Application Note:

This requirement applies to the Group Temporal Key (GTK) that is generated by the TOE for use in broadcast and multicast messages to clients to which it's connected. 802.11-2007 specifies the format for the transfer as well as the fact that it must be wrapped by the AES Key Wrap method specified in RFC 3394.

Assurance Activity:

The evaluator shall check the TSS to ensure that it describes how the GTK is wrapped prior to be distributed using the AES implementation specified in this PP, and also how the GTKs are distributed when multiple clients connect to the TOE. The evaluator shall also perform the following test:

- Test 1: The evaluator shall successfully connect multiple clients to the TOE. As the clients are connected, the evaluator shall observe that the GTK is not transmitted in the clear between the client and the TOE.
- Test 2: The evaluator shall cause a broadcast message to be sent to all clients connected to the TOE. The evaluator shall ensure the message is encrypted and cannot be read.
- Test 3: The evaluator shall create at least two multicast groups among a subset of clients connected to the TOE, each consisting of at least two clients but less than all of the clients connected to the TOE. Some (but not all) of the clients shall be in both groups. The evaluator shall ensure that GTKs established are sent to the participating clients and cannot be determined from the traffic flowing between the clients and the TOE.
- Test 4: The evaluator shall cause a multicast message to be sent to the clients in each multicast group connected to the TOE. The evaluator shall ensure each message is encrypted and cannot be read.

6.1.2.5 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

PP Application Note:

Any security related information (such as keys, authentication data, and passwords) must be zeroized when no longer in use to prevent the disclosure or modification of security critical data.

The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

Assurance Activity

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and critical security parameters used to generate keys; when they are zeroized

(for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the type of the memory or storage in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").

6.1.2.6 FCS_COP.1(1) Cryptographic Operation (Data Encryption/Decryption)

FCS_COP.1.1(1)

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in **CBC** and cryptographic key sizes 128-bits, 256-bits, and 192 bits that meets the following:

- FIPS PUB 197, “Advanced Encryption Standard (AES)”
- NIST SP 800-38A

PP Application Note:

For the assignment, the ST author should choose the mode or modes in which AES operates. For the first selection, the ST author should choose the key sizes that are supported by this functionality. For the second selection, the ST author should choose the standards that describe the modes specified in the assignment.

Note that this requirement does not apply to wireless traffic encryption. Requirement FCS_COP.1(5) defines the mode, key size and standards that are used for wireless WPA2 encryption/decryption.

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", “The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

6.1.2.7 FCS_COP.1(2) Cryptographic Operations (Cryptographic Signature)

FCS_COP.1.1(2)

The TSF shall perform cryptographic signature services in accordance with a:

- RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater
- Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater

that meets the following:

- FIPS PUB 186-3, “Digital Signature Standard”

PP Application Note:

As the preferred approach for cryptographic signature, elliptic curves will be required in future publications of this PP.

The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of this PP.

Assurance Activity

The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSA2VS), and "The RSA Validation System" (RSAVS (for 186-2) or RSA2VS (for 186-3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.8 FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing)

FCS_COP.1.1(3)

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-1, SHA-256, SHA-384 and message digest sizes 160, 256, 384 bits that meet the following: FIPS Pub 180-3, "Secure Hash Standard."

PP Application Note:

The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.

Assurance Activity:

The evaluator shall use "The Secure Hash Algorithm Validation System (SHAVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.9 FCS_COP.1(4) Cryptographic Operation (Keyed Hash Message Authentication)

FCS_COP.1.1(4)

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- SHA-1, SHA-256, SHA-384, key size **160, 256, 384 bits**, and message digest size of 160, 256, 384 bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

PP Application Note:

The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if HMAC-SHA-256 is chosen, then the only valid message digest size selection would be 256 bits.

The message digest size above corresponds to the underlying hash algorithm used. Note that truncating the output of the HMAC following the hash calculation is an appropriate step in a variety of applications. This does not invalidate compliance with this requirement, however, the ST should state that truncation is performed, the size of the final output, and the standard to which this truncation complies.

Assurance Activity:

The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.10 FCS_COP.1(5) Cryptographic Operation (WPA2 Data Encryption/Decryption)

FCS_COP.1.1(5)

The TSF shall perform encryption and decryption in accordance with the specified cryptographic algorithm AES CCMP and cryptographic key size of 128 bits that meet the following: FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2007.

PP Application Note:

Note that to comply with IEEE 802.11-2007, AES CCMP (which uses AES in CCM as specified in SP 800-38C) with cryptographic key size of 128 bits must be implemented. In the future, as this standard is updated and new cryptographic modes are reviewed and approved by NIST, this requirement may include requirements for additional/new cryptographic modes and key sizes.

Assurance Activity:

The evaluator shall use tests from "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)" as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

Additionally, the evaluator shall use tests from the IEEE 802.11-02/362r6 document “Proposed Test vectors for IEEE 802.11 TGi”, dated September 10, 2002, Section 2.1 AES-CCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.

6.1.2.11 FCS_IPSEC_EXT.1 Internet Protocol Security (IPsec) Communications

Assurance Activity:

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

- For each section of each applicable RFC listed for the FCS_IPSEC_EXT.1 elements, for all statements that are not "MUST" (for example, "MAY", "SHOULD", "SHOULD NOT", etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "SHOULD NOT" or "MUST NOT" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

- For each section of each RFC, any omission of functionality related to "MUST" or “SHOULD” statements shall be described;
- Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

The evaluator shall ensure the TSS identifies all servers/services that require or allow IPsec connections. The evaluators shall also ensure that when performing testing and analysis activities, the activities apply to all servers identified. The evaluators shall ensure that at least one instance of every type of server is used in at least one test during the testing activities to provide assurance that the identified communications can take place. The evaluators shall also ensure that the configuration information (including product and version numbers) for the non-TOE endpoints of these connections is recorded in the test report.

The evaluator shall also perform the following test for TOEs that implement IKEv2:

- Test 1 [conditional]: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 4306, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

FCS_IPSEC_EXT.1.1

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602), no other algorithms, and using IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, and no other RFCs for hash functions**;** IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and no other RFCs for hash functions for connections to the Authentication Server and **Syslog Server, and NTP Server.**

PP Application Note:

FCS_IPSEC_EXT.1 is supported at least for protection of the RADIUS communications between the WLAN Access System and an Authentication Server. The first selection is used to identify additional cryptographic algorithms supported. Either IKEv1 or IKEv2 support must be provided, although conformant TOEs can provide both; the second selection is used to make this choice. For IKEv1, the requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. RFC 4868 identifies additional hash functions for use with both IKEv1 and IKEv2; if these functions are implemented, the third (for IKEv1) and fourth (for IKEv2) selection can be used. The last selection/assignment is used to specify other servers/services (e.g., an audit server) the TOE communicates with whose communications are protected by IPsec.

FCS_IPSEC_EXT.1.2

The TSF shall ensure that only ESP confidentiality and integrity security service is used.

Assurance Activity:

The evaluator shall examine the TSS to verify that it describes how the "confidentiality only" ESP security service is disabled. The evaluator shall also examine the operational guidance to determine that it describes any configuration necessary to ensure negotiation of "confidentiality only" security service for ESP is disabled, and that an advisory is present indicating that tunnel mode is the preferred ESP mode since it protects the entire packet.

- Test 1: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using ESP using the "confidentiality only" security service. This attempt should fail. The evaluator shall then establish a connection using ESP using the confidentiality and integrity security service.

FCS_IPSEC_EXT.1.3

The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Assurance Activity:

The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. If this requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.

FCS_IPSEC_EXT.1.4

The TSF shall ensure that IKEv1 SA lifetimes are able to be limited by number of kilobytes/number of bytes, length of time, where the time can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs; IKEv2 SA lifetimes can be configured by an administrator based on number of packets/number of bytes; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.[4]

PP Application Note:

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in the first requirement. The IKEv1 requirement can be accomplished either by providing Authorized Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE), or by "hard coding" the limits in the implementation. For IKEv2, there are no hardcoded limits, but in this case it is required than an administrator be able to configure the values. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the administrative guidance generated for AGD_OPE. It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignments in FCS_IPSEC_EXT.1.5 and FCS_IPSEC_EXT.1.6 may contain multiple values. For each DH group supported, the ST author consults Table 2 in 800-57 to determine the "bits of security" associated with the DH group. Each unique value is then used to fill in the assignment (for 1.5 they are doubled; for 1.6 they are inserted directly into the assignment). For example, suppose the implementation support DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192. For

[4] Modified per TD0021

FCS_IPSEC_EXT.1.5, then, the assignment would read "[224, 384]" and for FCS_IPSEC_EXT.1.6 it would read "[112,192]" (although in this case the requirement should probably be refined so that it makes sense mathematically).

Assurance Activity:

If IKEv1 requirements are selected, the evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established. If they are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance. For IKEv2 requirements, the evaluator verifies that the values can be configured and that the instructions for doing so are located in the operational guidance. The evaluator also performs the following tests, depending on whether IKEv1, IKEv2, or both are configured:

- Test 1 (IKEv1): The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.
- Test 2 (IKEv1): The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.
- Test 3 (IKEv1 and v2): The evaluator shall configure a maximum lifetime in terms of the # of packets allowed; this may be a hard-coded value for IKEv1, otherwise, the evaluator follows the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets through this SA is exceeded, the connection is closed.
- Test 4 (IKEv2): The evaluator shall configure a time-based maximum lifetime for an SA, and then establish the SA. The evaluator shall observe that this SA is closed or renegotiated in the established time.

FCS_IPSEC_EXT.1.5

The TSF shall generate the secret value x used in the IKE Diffie Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least **224/256/384**[5] bits.

Assurance Activity:

The evaluator shall check to ensure that, for each DH group supported by the TSF, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.5) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

FCS_IPSEC_EXT.1.6

The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^**256**.

Assurance Activity:

[5] Bit Values correspond to DH Group 14:112/ DH Group 19:128/ DH Group 20:192/

(See FCS_IPSEC_EXT.1.5 Assurance Activities)

FCS_IPSEC_EXT.1.7

The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP) and 19 (256-bit Random ECP), 20 (384-bit Random ECP)**.**

PP Application Note:

The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. In future versions of this PP, DH Groups 19 (256-bit Random ECP) and 20 (384-bit Random ECP) will be required. It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS_CKM.1(2).

Assurance Activity:

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer. The evaluator shall also perform the following test:

- Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

FCS_IPSEC_EXT.1.8

The TSF shall ensure that all IKE protocols implement peer authentication using Pre-shared Keys and ECDSA that use X.509v3 certificates that conform to RFC 4945.

PP Application Note:

Pre-shared keys and at least one public-key-based Peer Authentication method are required for conformant TOEs; one or more of the public key schemes is chosen by the ST Author to reflect what is implemented by the TOE. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).

Assurance Activity:

The evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key. The evaluator shall also perform the following test:

- Test 1: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection between two peers. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.

The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the algorithm or algorithms specified in the

selection. As part of the assurance activity for FCS_IPSEC_EXT.1.1, required and optional elements of RFC 4945 shall be documented. The evaluator shall also perform the following tests:

- Test 1: For each supported algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved.
- Test 2: For each supported identification payload (from RFC 4945), the evaluator shall test that peer authentication can be successfully achieved.
- Test 3: The evaluator shall devise a test that demonstrates that a corrupt or invalid certification path for a certificate will be detected during IKE peer authentication and will result in a connection not being established.
- Test 4: The evaluator shall devise a test that demonstrates that a certificate that has been revoked through a CRL will be detected during IKE peer authentication and will result in a connection not being established.

FCS_IPSEC_EXT.1.9

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection.

PP Application Note:

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for a TOE to allow this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the OPE documentation) must enable this functionality.

Assurance Activity[6]:

The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation. The evaluator shall also perform the following tests:

- Test 1: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.
- Test 2: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

[6] Due to an apparent typo in the PP, the ST includes the Assurance Activities written for FCS_IPSEC_EXT.1.10, which does not appear in the PP.

6.1.2.12 FCS_TLS_EXT.1 Transport Layer Security (TLS)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols: TLS 1.0 (RFC 2246)[7] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

Optional Ciphersuites:

- None

PP Application Note:

The ST author must make the appropriate selections and assignments to reflect the TLS implementation. The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.

The ciphersuites to be used in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.

The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. Future publications of this PP will require support for TLS 1.2 (RFC 5246). In addition, future publications of this PP will require that the TOE offer a means to deny all connection attempts using specified older versions of the SSL/TLS protocol.

Assurance Activity:

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

- For each section of each applicable RFC listed for the FCS_TLS_EXT.1 elements, for all statements that are not "MUST" (for example, "MAY", "SHOULD", "SHOULD NOT", etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "SHOULD NOT" or "MUST NOT" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each section of each RFC, any omission of functionality related to "MUST" or "SHOULD" statements shall be described;

[7] Due to an apparent typo in the PP, the RFC number in the SFR has bene updated.

- Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe (on the wire) the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

6.1.2.13 FCS_SSH_EXT.1 Secure Shell (SSH)

Assurance Activity:

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

- For each section of each applicable RFC listed for the FCS_SSH_EXT.1 elements, for all statements that are not "MUST" (for example, "MAY", "SHOULD", "SHOULD NOT", etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "SHOULD NOT" or "MUST NOT" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each section of each RFC, any omission of functionality related to "MUST" or “SHOULD” statements shall be described;
- Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

FCS_SSH_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254 and no other RFCs.[8]

PP Application Note:

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted).[9]

[8] Updated based on TD009

[9] Updated based on TD009

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH connection be rekeyed after no more than 2^{28} packets have been transmitted using that key.

Assurance Activity:

The evaluator shall examine the TSS to ensure that it specifies that the TOE rekeys an SSH connection before more than 2^{28} packets have been sent with a given key. If this effect is achieved by configuration of the TOE, then the evaluator shall examine the operational guidance to ensure that it contains instructions on setting the appropriate values.

FCS_SSH_EXT.1.3

The TSF shall ensure that the SSH protocol implements a timeout period for authentication as defined in RFC 4252 of **60 seconds**, and provide a limit to the number of failed authentication attempts a client may perform in a single session to **3** attempts**.**

PP Application Note:

In the first assignment, the ST author should insert the timeout period (e.g., "10 minutes") from the initiation of authentication session after which the session should timeout if authentication has been unsuccessful. In the second assignment, the maximum number of failed authentication attempts is specified. The RFC indicates the server should drop the session after this number of failed attempts.

Assurance Activity:

The evaluator shall check to ensure that the TSS specifies the timeout period and the method for dropping a session connection after the number of failed authentication attempts specified in the requirement. If these values are configurable and may be specified by the administrator, the evaluator shall check the operational guidance to ensure that it contains instructions for configuring these values. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall demonstrate that taking longer than the timeout period to authenticate to the TOE results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If the timeout period is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different periods in order to ensure that the mechanism works as specified.
- Test 2: The evaluator shall demonstrate that performing a number of failed SSH authentication attempts equal to the value specified in the requirement results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If this number is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different limits (e.g., 3 attempts and 5 attempts) in order to ensure that the mechanism works as specified.

FCS_SSH_EXT.1.4

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

Assurance Activity:

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.7, and ensure that

password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.
- Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

FCS_SSH_EXT.1.5

The TSF shall ensure that, as described in RFC 4253, packets greater than **32768** bytes in an SSH transport connection are dropped.

PP Application Note:

RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

Assurance Activity:

The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

- Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSH_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, **no other encryption algorithms.**

PP Application Note:

In subsequent publications of this PP, it is likely that AES-GCM will be required and CBC will become optional. In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of a protocol to satisfy the intent of the test.

FCS_SSH_EXT.1.7

The TSF shall ensure that the SSH transport implementation uses SSH_RSA and no other public key algorithms as its public key algorithm(s).

PP Application Note:

RFC 4253 specifies required and allowable public key algorithms. This requirement makes SSH-RSA "required" and allows two others to be claimed in the ST. The ST author should make the appropriate selection, selecting "no other public key algorithms" if only SSH_RSA is implemented.

Assurance Activity:

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.8

The TSF shall ensure that the data integrity algorithm used in the SSH transport connection is hmac-sha1 and hmac-sha1-96.

PP Application Note:

As per the RFC, HMAC-SHA1 is required, but there are additional integrity algorithms that are allowed. The ST author chooses the algorithm(s) implemented by the TOE; if there are no additional algorithms, then that should be selected.

Assurance Activity:

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

FCS_SSH_EXT.1.9

The TSF shall ensure that diffie-hellman-group14-sha1 and ecdh-sha2-NISTP256, ecdh-sha2-NISTP384 are the only allowed key exchange method used for the SSH protocol.[10]

Assurance Activity:

The evaluator shall ensure that operational guidance contains configuration information that will allow an authorized administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14. If this capability is "hard-coded" into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

- Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method, the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.

[10] Updated based on TD0009

6.1.2.14 FCS_HTTPS_EXT.1 HTTP Security (HTTPS)

PP Application Note:

The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.

FCS_HTTPS_EXT.1.1

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

Assurance Activity:

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

- For each section of each applicable RFC listed for the FCS_HTTPS_EXT.1 elements, for all statements that are not "MUST" (for example, "MAY", "SHOULD", "SHOULD NOT", etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "SHOULD NOT" or "MUST NOT" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each section of each RFC, any omission of functionality related to "MUST" or “SHOULD” statements shall be described;
- Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

Assurance Activity:

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. administrator authentication which may be done at a different level of the processing stack. Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

6.1.2.15 FCS_RBG_EXT.1 Extended Cryptographic Operation (Random Bit Generation)

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with NIST Special Publication 800-90 using HMAC_DRBG (any) seeded by an entropy source that accumulated entropy from at least one independent TSF-hardware based noise sources.

FCS_RBG_EXT.1.2

Refinement: The deterministic RBG shall be seeded with a minimum of 256 bits of entropy at least equal to the greatest ***~~bit length~~ security strength*** of the keys and authorization factors ***and hashes*** that it will generate.[11]

PP Application Note:

NIST Special Pub 800-90B describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of the NDPP.

For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90B or 140-2 Annex C).

SP 800-90B contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90B is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224,SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed. While any of the curves defined in 800-90B are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.

Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

In the future, most of the requirements described in A Method for Entropy Source Testing: Requirements and Test Suite Description will be required by this PP. The follow Assurance Activities currently reflect only that subset of activities that are required.

Assurance Activity:

The evaluator shall review the TSS section to determine the version number of the product containing the RBG(s) used in the TOE. The evaluator shall also confirm that the TSS describes the hardware-based noise source from which entropy is gathered, and further confirm that this noise source is located on the TOE12. The evaluator will further verify that all of the underlying functions and parameters used in the RBG are listed in the TSS.

The evaluator shall verify that the TSS contains a description of the RBG model, including the method for obtaining entropy input, as well as identifying the entropy source(s) used, how entropy is produced/gathered from each source, and how much entropy is produced by each entropy source. The evaluator shall also ensure that the TSS describes the entropy source health tests, a rationale for why the health tests are sufficient to determine the health of the entropy sources, and known modes of entropy source failure. Finally, the evaluator shall ensure that the TSS contains a description of the RBG

[11] This refinement is for consistency with the NDPP v1.1 Errata #2 as requested by validators.

[12] This was changed from USB Flash Drive to TOE per TRRT response.

outputs in terms of the independence of the output and variance with time and/or environmental conditions.

Regardless of the standard to which the RBG is claiming conformance, the evaluator perform the following test:

- Test 1: The evaluator shall determine an entropy estimate for each entropy source by using the Entropy Source Test Suite. The evaluator shall ensure that the TSS includes an entropy estimate that is the minimum of all results obtained from all entropy sources.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to FIPS 140-2, Annex C

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluator ensures that the values returned by the TSF match the expected values.

The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.

Implementations Conforming to NIST Special Publication 800-90

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input,

nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

- Entropy input: the length of the entropy input value must equal the seed length.
- Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.
- Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is supported, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.
- Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects.

PP Application Note:

This requirement ensures, for example, that protocol data units (PDUs) are not padded with residual information such as cryptographic key material. The ST author uses the selection to specify when previous information is made unavailable.

Assurance Activity:

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer of successive unsuccessful authentication attempts occur related to administrators attempting to authenticate remotely.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall prevent the offending remote administrator from successfully authenticating until **account unlock action** is taken by a local Administrator or prevent the offending remote administrator from successfully authenticating until an Administrator defined time period has elapsed.

PP Application Note:

This requirement does not apply to an administrator at the local console, since it does not make sense to lock a local administrator's account in this fashion. This could be addressed by (for example) requiring a separate account for local administrators or having the authentication mechanism implementation distinguish local and remote login attempts. The "action" taken by a local administrator is implementation specific and would be defined in the administrator guidance (for example, lockout reset or password reset). The ST author chooses one of the selections for handling of authentication failures depending on how the TOE has implemented this handler.

Assurance Activity:

The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

The evaluator shall also examine the operational guidance to ensure that instructions for configuring the number of successive unsuccessful authentication attempts (1.1) and time period (1.2, if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

The evaluator shall perform the following tests for IPsec, and for each other method by which remote administrators access the TOE (e.g., TLS, SSH):

- Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE. The evaluator shall test that once the limit is reached, attempts with valid credentials are not successful. For each action specified by the requirement, the evaluator shall show that following the operational guidance and performing each action to allow the remote administrator access are successful.
- Test 2: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE and a time period after which valid logins will be allowed for a remote administrator. After exceeding the specified number of invalid login attempts and showing that valid login is not possible, the evaluator shall show that waiting for the interval defined by the time period before another access attempt will result in the ability for the remote administrator to successfully log on using valid credentials.

6.1.4.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1

The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "$", "#", "$", "%", "^", "&", "*", "(", ")";
2. Minimum password length shall settable by the Security Administrator, and support passwords of 8 characters or greater;
3. Password composition rules specifying the types and number of required characters that comprise the password shall be settable by the Administrator.
4. Passwords shall have a maximum lifetime, configurable by the Authorized Administrator.
5. New passwords must contain a minimum of 4 character changes from the previous password.[13]

PP Application Note:

Note that it is not necessary to store a plaintext version of the password in order to determine that at least 4 characters have changed, since FIA_UAU.6 requires re-authentication when changing the password.

"Administrative passwords" refers to passwords used by administrators at the local console or over protocols that support passwords, such as SSH and HTTPS.

The intent of Item 3 above is that an Authorized Administrator is able to specify, for example, that passwords contain at least 1 upper case letter, 1 lower case letter, 1 numeric character, and 1 special character, and the TOE enforces this restriction. "Types" refers to all of the types listed in Item 1 in this element.

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

- Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.
- Test 2: The evaluator shall ensure that the operational guidance contains instructions on setting he maximum password lifetime. The evaluator shall then configure this lifetime to several values, and ensure that it is enforced for each of those values.
- Test 3: The evaluator shall test that a minimum of 4 character changes from previous passwords is enforced. This shall be done for more than one password.

6.1.4.3 FIA_PSK_EXT.1Extended: Pre-Shared Key Composition

FIA_PSK_EXT.1.1

The TSF shall be able to use pre-shared keys for IPsec and **WPA2_PSK**.

[13] Modified per TD0002

FIA_PSK_EXT.1.2(1)

The TSF shall be able to accept text-based pre-shared keys ***for IPsec*** that:

- are 22 characters and **16 to 128 characters**;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

FIA_PSK_EXT.1.2(2)

The TSF shall be able to accept text-based pre-shared keys ***for WPA2_PSK*** that:

- are 22 characters and **8 to 63 characters**;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

FIA_PSK_EXT.1.3

The TSF shall condition the text-based pre-shared keys by using SHA-256.

FIA_PSK_EXT.1.4

The TSF shall be able to accept, generate using the random bit generator specified in FCS_RBG_EXT.1 bit-based pre-shared keys.

PP Application Note:

In the first selection, if other protocols can use pre-shared keys, they should be listed in the assignment as well; otherwise "no other protocols" should be chosen. The intent of this requirement is that all protocols will support both text-based and bit-based pre-shared keys.

For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.

In the selection for FIA_PSK_EXT.1.3, the ST author selects or fills in the method by which the text string entered by the administrator is "conditioned" into the bit string used as the key. This can be done by using one of the specified hash functions, or some other method through the assignment statement.

For FIA_PSK_EXT.1.4, the ST author specifies whether the TSF merely accepts bit-based pre-shared keys, or is capable of generating them. If it generates them, the requirement specified that they must be generated using the RBG provided by the TOE.

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

The evaluator shall examine the TSS to ensure that it identifies all protocols that allow both text-based and bit-based pre-shared keys, and states that text-based pre-shared keys of 22 characters are supported. For each protocol identified by the requirement, the evaluator shall confirm that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key

sequence entered by the user (e.g., ASCII representation) to the bit string used by the protocol, and that this conditioning is consistent with the last selection in the FIA_PSK_EXT.1.3 requirement.

The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case.

- Test 1: The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates that a successful protocol negotiation can be performed with the key.
- Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length, and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.
- Test 3 [conditional]: If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
- Test 4 [conditional]: If the TOE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key. FIA_UIA_EXT.1 User Identification and Authentication

6.1.4.4 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

The TSF shall allow responses to the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- **Receive and send MVP (Mesh Viewer Protocol) packets every 30 seconds on port 4949.**

FIA_UIA_EXT.1.2

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

PP Application Note:

This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are

some available (perhaps ICMP echo) these should be listed in the assignment statement; otherwise "no other actions" should be selected.

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection "counts" as initiating the identification and authentication process.

Assurance Activity:

The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

- Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.
- Test 2: The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
- Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

6.1.4.5 FIA_UAU_EXT.5 Password-based Authentication Mechanism

FIA_UAU_EXT.5.1

The TSF shall provide a local password-based authentication mechanism **and external RADIUS** to perform administrative user authentication.

FIA_UAU_EXT.5.2

The TSF shall ensure that administrative users with expired passwords are required to create a new password after correctly entering the expired password.

PP Application Note:

This requirement only applies to the local administrator login, and essentially requires that a password-based mechanism exists on the TOE for this purpose. The ST author can fill in the assignment with any other supported authentication mechanisms (such as an authentication server) for administrative users that are not local. If no external authentication mechanisms for administrative users are supported, the ST author should choose "none" in the selection.

Assurance Activities:

Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1

6.1.4.6 FIA_UAU.6 Re-authenticating

FIA_UAU.6.1

The TSF shall re-authenticate the administrative user under the conditions: when the user changes their password, no other conditions.

Assurance Activities:

The evaluator shall perform the following test for each of the conditions specified in the requirement:

- Test 1: The evaluator shall attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authentication is required.

6.1.4.7 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

PP Application Note:

"Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

Assurance Activity:

The evaluator shall perform the following test for each method of local login allowed:

- Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

6.1.4.8 FIA_8021X_EXT.1 802.1X Port Access Entity (Authenticator) Authentication

FIA_8021X_EXT.1.1

The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the "Authenticator" role.

FIA_8021X_EXT.1.2

The TSF shall support communications to a RADIUS authentication server conforming to RFCs 2865 and 3579.

FIA_8021X_EXT.1.3

The TSF shall ensure that no access to its 802.1X controlled port is given to the wireless client prior to successful completion of this authentication exchange.

PP Application Note:

This requirement covers the TOE's role as the authenticator in an 802.1X authentication exchange. If the exchange is completed successfully, the TOE will obtain the PMK from the RADIUS server and perform the 4-way handshake with the wireless client (supplicant) to begin 802.11 communications.

As indicated previously, there are at least three communication paths present during the exchange; two with the TOE as an endpoint and one with TOE acting as a transfer point only. The TOE establishes an EAP over LAN (EAPOL) connection with the wireless client as specified in 802.1X-2007. The TOE also establishes (or has established) a RADIUS protocol connection (which is tunneled inside of an IPsec connection) with the RADIUS server. The wireless client and RADIUS server establish an EAP-TLS session (RFC 5216); in this transaction the TOE merely takes the EAP-TLS packets from its EAPOL/RADIUS endpoint and transfers them to the other endpoint. Because the specific authentication method (TLS in this case) is opaque to the TOE, there are no requirements with respect to RFC 5126 in this PP. However, the base RADIUS protocol (2865) has an update (3579) that will need to be addressed in the implementation and assurance activities. Additionally, RFC 5080 contains implementation issues that will need to be addressed by developers, but which levy no new requirements.

The point of performing 802.1X authentication is to provide access to the network (assuming the authentication was successful and that all 802.11 negotiations are performed successfully); in the terminology of 802.1X, this means the wireless client has access to the "controlled port" maintained by the TOE.

Assurance Activity:

In order to show that the TSF implements the 802.1X-2010 standard correctly, the evaluator shall ensure that the TSS contains the following information:

- the sections (clauses) of the standard that the TOE implements;
- For each identified section, any options allowed by the standards are specified; and
- For each identified section, any non-conformance is identified and described, including a justification for the non-conformance.

Because the connection to the RADIUS server will be contained in an IPsec tunnel (FCS_IPSEC_EXT.1), the security mechanisms detailed in the RFCs identified in the requirement are not relied on to provide protection for these communications. Consequently, no extensive analysis of the RFCs is required. However, the evaluator shall ensure that the TSS describes the measures (documentation, testing) that are taken by the product developer to ensure that the TOE conforms to the RFCs listed in this requirement.

The evaluator shall also perform the following tests:

- Test 1: The evaluator shall demonstrate that a wireless client has no access to the test network. After successfully authenticating with a RADIUS server through the TOE, the evaluator shall demonstrate that the wireless client does have access to the test network.

- Test 2: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.
- Test 3: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid RADIUS certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.

It should be noted that tests 2 and 3 above are not tests that "EAP-TLS works", although that is a by-product of the test. The test is actually that a failed authentication (under two failure modes) results in denial of access to the network, which is the 3rd element of this component.

6.1.4.9 FIA_X509_EXT.1 Extended: X.509 Certificates

FIA_X509_EXT.1.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and TLS connections.

FIA_X509_EXT.1.2

The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.1.3

The TSF shall provide the capability for authenticated Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this ***ST ~~PP~~***.

PP Application Note:

For FIA_X509_EXT.1.1, the ST author should select the protocols that are used to implement administrative connectivity that also use certificates for authentication. It should be noted that RFC 5280 defines certificate validation and certification path validation requirements that must be implemented by the TOE as per this requirement.

Depending on the protocols selected, there may be additional protocol-specific certificate-related requirements (and associated assurance activities) specified (for instance, RFC 4945 for IPsec). These additional requirements are specified in the requirements associated with that protocol.

FIA_X509_EXT.1.2 applies to certificates that are used and processed by the TSF. Certificates that are used and process by other components in the Operational Environment (e.g., the RADIUS server) are not intended to be covered by this element.

Assurance Activity:

In order to show that the TSF supports the use of X.509v3 certificates according to the RFC 5280, the evaluator shall ensure that the TSS describes the following information:

- For each section of RFC 5280, any statement that is not "MUST" (for example, "MAY", "SHOULD", "SHOULD NOT", etc.) shall be described so that the reader can determine whether the TOE implements that specific part of the standard;
- For each section of RFC 5280, any non-conformance to "MUST" or “SHOULD" statements shall be described;

- Any TOE-specific extensions or processing that is not included in the standard that may impact the security requirements the TOE is to enforce shall be described.

Additionally, the evaluator shall devise tests that show that the TOE processes certificates that conform to the implementation described in the TSS; are able to form a certification path as specified in the standard and in the TSS; and are able to validate certificates as specified in the standard (certification path validation including CRL processing). This testing shall be described in the team test plan.

It should be noted that future versions of this PP will have more explicit testing requirements for a TOE's certificate handling capability. Additionally, protocol-specific certificate handling testing will need to be performed and can be combined with the testing required by this assurance activity.

The TSS shall describe all certificate stores implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access.

The evaluator shall perform the following tests for each function in the system that requires the use of certificates:

- Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails

6.1.5 Security Management (FMT)

6.1.5.1 FMT_MOF.1 Management of Security Functions Behavior

FMT_MOF.1.1

Refinement: The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this ***~~PP~~ ST*** to the Authorized Administrator.

PP Application Note:

The only human users of the TOE are administrative users; therefore, this requirement is present to underscore the fact that non-administrative users will not be able to manipulate the mechanisms of the TOE used to implement the security requirements of the PP. These capabilities explicitly cover functions implemented in the TOE dealing with adding TOE components to the network and structuring them from a management or redundancy standpoint.

Assurance Activity:

The evaluator shall review the operational guidance to determine that each of the functions implemented in response to the requirements of this PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall include in this list of functions to be examined those mechanisms dealing with adding additional instances of a TOE to a configuration, and configuration of the multiple TOE instances into a management hierarchy and/or redundant architecture. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance, those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the configuration of the system through this interface is disallowed for non-administrative users.

6.1.5.2 FMT_MTD.1(1) Management of TSF Data (General TSF data)

FMT_MTD.1.1(1)

The TSF shall restrict the ability to manage the TSF data to the Authorized Administrators.

PP Application Note:

The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This requirement is intended to be the "default" requirement for management of TSF data; other iterations of FMT_MTD should place different restrictions or operations available on the specifically-identified TSF data. TSF data includes cryptographic information as well; managing these data would include the association of a cryptographic protocol with an interface, for instance.

Assurance Activity:

Since administrative functions manipulate the TSF data, the analysis performed by the evaluators in the Assurance Activity for FMT_MOF.1 will demonstrate that this requirement is met

6.1.5.3 FMT_MTD.1(2) Management of TSF Data (Reading of Authentication Data)

FMT_MTD.1.1(2)

The TSF prevent the reading of password-based authentication data.

PP Application Note:

The intent of the requirement is that no user or administrator be able to read the raw authentication data (such as an unencrypted password) through "normal" interfaces if the reading of such data could lead to someone impersonating that user. An all-powerful administrator of course could directly read memory or do a raw read of the file system to capture a password but is trusted not to do so.

Assurance Activity:

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and how they are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If passwords or other authentication data are not stored in plaintext, the TSS shall describe how the passwords are protected and how they are able to be used (e.g., administrator-entered passphrase).

6.1.5.4 FMT_MTD.1(3) Management of TSF Data (for reading of all symmetric keys)

FMT_MTD.1.1(3)

The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

PP Application Note:

The intent of the requirement is that no user or administrator be able to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While an authorized administrator of course could directly read memory to view these keys, they are trusted not to do so

Assurance Activity:

The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed

specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

6.1.5.5 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:

- Ability to configure the list of TOE services available before an entity is identified and authenticated, as specified in FIA_UIA.1, respectively,
- Ability to configure the cryptographic functionality,
- Ability to update the TOE, and to verify the updates using the digital signature capability (FCS_COP.1(2)) and no other functions.
- Ability to configure the TOE advisory notice and consent warning message regarding unauthorized use of the TOE,
- Ability to configure all security management functions identified in other sections of this ~~***PP***~~ ***ST***.

PP Application Note:

The security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MOF, FMT_MSA, FMT_MTD, FMT_REV, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards.

Assurance Activity:

This requirement merely ensures that the mechanisms called for in other requirements are actually instantiated in the TOE; therefore, verification that these mechanisms exist and work in a manner consistent with the other requirements is provided through the Assurance Activities associated with those other requirements.

6.1.5.6 FMT_SMR.1 Security Management Roles

FMT_SMR.1.1

The TSF shall maintain the roles:

- Authorized Administrator
- No other roles

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

FMT_SMR.1.3

The TSF shall ensure that the conditions

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;
- The ability to remotely administer the TOE remotely from a wireless client shall be disabled by default .

are satisfied.

PP Application Note:

FMT_SMR.1.2 requires that user accounts be associated with only one role. However, note that multiple users may have the same role, and the TOE is not required to restrict roles to a single person.

FMT_SMR.1.3 requires that an authorized administrator be able to administer the TOE through the local console and through a remote mechanism (IPsec, SSH, TLS/HTTPS). For multiple component TOEs, only the TOE components providing the management control and configuration of the other TOE components require a local administration interface.

Assurance Activity:

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

The evaluator shall also perform the following test:

- Test 1: The evaluator shall demonstrate that after configuring the TOE for first use from the operational guidance, it is possible to establish an administrative session with the TOE on the "wired" portion of the device. They shall then demonstrate that an identically configured wireless client that can successfully connect to the TOE cannot be used to perform administration.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_FLS.1 Fail Secure

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur: failure of the power-on self-tests.

PP Application Note:

The intent of this requirement is to express the fail secure capabilities that the TOE possesses. This means that the TOE must be able to attain a secure/safe state when any of the identified failures occurs. If the TOE should encounter a failure in the middle of a critical operation, the TOE should not just quit operating leaving key material and user data unprotected.

Assurance Activity:

The evaluator shall review the TSS section to determine that the TOE's implementation of the fail secure functionality is documented. The evaluator shall first examine the TSS section to ensure that all failure modes specified in the ST are described. The evaluator shall then ensure that the TOE will attain a secure state after inserting each specified failure mode type. The evaluator shall review the TSS to determine that the definition of secure state is defined and is suitable to ensure protection of key material and user data.

6.1.6.2 FPT_RPL.1 Replay Detection

FPT_RPL.1.1

The TSF shall detect replay for the following entities: network packets terminated at the TOE.

FPT_RPL.1.2

The TSF shall perform: reject the data when replay is detected.

PP Application Note:

Receiving multiple network packets due to network congestion or lost packet acknowledgments is not considered a replay attack. The intent of this requirement is to ensure that any communications of a trusted nature (administrator to TOE, IT entity to TOE, TOE to TOE) are covered by the element and cannot be replayed.

6.1.6.3 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1

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

6.1.6.4 FPT_TST_EXT.1 Extended TSF Testing

FPT_TST_EXT.1.1

The TSF shall run a suite of self tests during the initial start-up (on power on) to demonstrate the correct operation of the TSF.

~~**FPT_TST_EXT.1.2**~~

~~The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).~~[14]

Assurance Activities:

The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

~~The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution, which includes the generation and protection of the "check value" used to ensure integrity as well as the verification step. This description shall also cover the digital signature service used in performing these functions. The evaluator also checks the operational guidance to ensure that any actions required by the administrator to initialize or operate this functionality are present.~~

[14] Removed per TD0022

~~The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases. The evaluator shall perform the following tests:~~

- ~~Test 1: Following the operational guidance, the evaluator shall initialize the integrity protection system. The evaluator shall perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors.~~
- ~~Test 2: The evaluator modifies the TSF executable, and causes that executable to be loaded by the TSF. The evaluator observes that an integrity violation is triggered (care must be taken so that the integrity violation is determined to be the cause of the failure to load the module, and not the fact that the module was modified so that it was rendered unable to run because its format was corrupt).~~[15]

6.1.6.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2

The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and no other functions prior to installing those updates.

PP Application Note:

The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2). The published hash referenced is generated by one of the functions specified in FCS_COP.1(3).

Assurance Activity:

Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:

- Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator

[15] Removed per TD0022

performs the version verification activity again to verify the version correctly corresponds to that of the update.

- Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. ~~***The evaluator verifies that the TOE rejects the update.***~~ ***The evaluator verifies that the TOE either rejects the update without intervention or detects that the update is illegitimate and allows the administrator to reject the update (as specified in the operational guidance).***[16]

6.1.7 Resource Utilization (FRU)

6.1.7.1 FRU_RSA.1 Maximum Quotas

FRU_RSA.1.1

The TSF shall enforce maximum quotas of the following resources: **connections to administrative interface through wireless connections limited to 127**, no other resources that defined group of users can use simultaneously.

PP Application Note:

At a minimum, compliant TOEs must impose quotas on exhaustible resources used to support the remote administrative interface; these are listed in the first assignment. Other resources that can be controlled (e.g., TCP connection resources) should be listed in the second assignment; if there are no other resources then the last item in the selection should be chosen. The second selection should be chosen to reflect the consumers of the resource that are to be controlled. The last selection is used to limit the timeframe associated with the use of the controlled resources (e.g., a quota on the number of TCP connection requests from a given IP address in 30 seconds).

Assurance Activity:

The evaluator shall examine the TSS to ensure that it identifies all resources controlled through the quota mechanism, and that this list contains those resources used to support the administrative interface. The evaluator shall ensure that the TSS describes how each resource is counted as "used" and how a maximum quota or use is determined, as well as the action taken when the quota is reached. The TSS shall also describe whether the quota is imposed on users or subjects (in this case TOE processes) and whether the quota imposed is for simultaneous use or cumulative use over a period of time. The evaluator shall examine the operational guidance to determine that it contains instructions for establishing quotas (if they are configurable), and describes any actions administrators can or should take in response to a quota being reached.

The evaluator shall also perform the following tests for each controlled resource:

- Test 1: The evaluator follows the operational guidance to configure quotas for the resource (if such a capability is provided). The evaluator then causes the resource quota to be reached, and observes that the action specified in the TSS occurs.

[16] This modification is required per TD26

6.1.8 TOE Access (FTA)

6.1.8.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions,

- terminate the session

after an Authorized Administrator-specified time period of inactivity.

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

6.1.8.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

The TSF shall terminate a remote interactive session after a Authorized Administrator-configurable time interval of session inactivity.

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

6.1.8.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.
- Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

6.1.8.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1

Before establishing an administrative user session the TSF shall be capable of displaying an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

PP Application Note:

This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.

Assurance Activity:

The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test:

- Test 1: The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

6.1.8.5 FTA_TSE.1 TOE Session Establishment

FTA_TSE.1.1

The TSF shall be able to deny establishment of a wireless client session based on location, time, day, **and no other attributes**.

PP Application Note:

The "location" can be specified in terms of a port number, IP address, subnet, VLAN, TOE interface, etc.

The assignment is to be used by the ST author to specify additional attributes on which denial of session establishment can be based.

Assurance Activity:

The evaluator shall examine the TSS to determine that all of the attributes on which a client session can be denied are specifically defined. The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS. The evaluator shall also perform the following test for each attribute:

- Test 1: The evaluator successfully establishes a client session with a wireless client. The evaluator then follows the operational guidance to configure the system so that that client's access is denied based on a specific value of the attribute. The evaluator shall then attempt to establish a session in contravention to the attribute setting (for instance, the location is denied based upon the client's IP address). The evaluator shall observe that the access attempt fails.

6.1.9 Trusted Path/Channels (FTP)

6.1.9.1 FTP_ITC.1 Inter-TSF-trusted channel

FTP_ITC.1.1

The TSF shall use 802.11-2007, IPsec, and no other protocols to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for **external authentication mechanisms (RADIUS), syslog, and NTP**.

PP Application Note:

The intent of the above requirement is to use a cryptographic protocol to protect all external communications with authorized IT entities that the TOE interacts with to perform its functions. 802.11-2007 is required for communications with wireless clients; IPsec is required at least for communications with the authentication server. If communications with other necessary authorized IT entities (NTP server, audit server), then they must use IPsec or one of the other listed protocols (SSH, TLS and TLS/HTTPS are allowed), and the ST author makes the appropriate selections. After the ST author has made the selections, they are to select the detailed requirements in Annex C corresponding to their selection to put in the ST.

While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

Assurance Activity:

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

- Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the

connections as described in the operational guidance and ensuring that communication is successful.

- Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.
- Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
- ~~Test 4: The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.~~[17]
- Test 5: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

6.1.9.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1

The TSF shall use SSH, TLS/HTTPS provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2

The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

PP Application Note:

This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.

Assurance Activity:

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

[17] Removed per TD0016

- Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
- Test 3: The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.
- ~~Test 4: The evaluator shall ensure, for each method of remote administration, modification of the channel data is detected by the TOE.~~[18]

Further assurance activities are associated with the specific protocols.

6.2 Security Assurance Requirements

This Security Target conformant with the assurance requirements specified in the PP. The CC Part 3 conformant security assurance requirements are listed in Table 9. The CC Part 3 extended assurance requirements are listed in Section 6.1 as "Assurance Activity" and Section 6.2.1.

Table 9: Assurance Requirements		
Assurance Class	Assurance Component	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Life-cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis

6.3 Security Requirements Rationale

6.3.1 Security Function Requirements Rationale

Table 10: TOE Security Functional Requirements Rationale satisfies the requirement to trace each SFR back to the security objectives for the TOE.

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
O.AUTH_COMM The TOE will provide a means to ensure users are not communicating with some other	FCS_IPSEC_EXT.1 FCS_TLS_EXT.1 FCS_SSH_EXT.1 FCS_HTTPS_EXT.1	FTP_ITC.1 and FTP_TRP.1 (and the supporting protocols 802.11-2007, FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1, and FCS_HTTPS_EXT.1) require the TOE provide a

[18] Removed per TD0016

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity.	FTP_ITC.1 FTP_TRP.1 FIA_8021X_EXT.1 FIA_UIA_EXT.1 FIA_PSK_EXT.1	mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification. FIA_8021X_EXT.1 provides the two-way authentication necessary to allow a wireless client access to the wired network, and serves as a part of the 802.11-2007 WPA2 protocol to establish the communication channel with the wireless client. FIA_UIA_EXT.1 requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. FIA_PSK_EXT.1 requires the TOE support the formation of strong pre-shared keys (either though a large character set for text-based pre-shared keys, or through generation by the TOE's (or an off-box) RBG function) that can be used to mutually authenticate the TOE and its communication partner. *Application Note: The ST author will modify the rationale to reflect the protocols that are implemented by the TOE.*
O.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.	FCS_CKM.1(1) FCS_CKM.1(2) FCS_CKM.2(1) FCS_CKM.2(2) FCS_CKM_EXT.4 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_COP.1(5) FCS_RBG_EXT.1 FIA_X509_EXT.1	FCS_CKM.1(1) and FCS_CKM.1(2) generate symmetric and asymmetric key, respectively. These keys are used by the AES encryption/decryption functionality specified in FCS_COP.1(5) and used for cryptographic signatures as specified in FCS_COP.1(2). FCS_CKM.2(1) and FCS_CKM.2(2) assures that the distribution method of cryptographic keys for wireless client communications are in accordance with a standard and do not get exposed. FCS_CKM_EXT.4 provides the functionality for ensuring key and key material is zeroized. This applies not only to key that resides in the TOE, but also to intermediate areas (physical memory, page files, memory dumps, etc.) where key material may appear. FCS_COP.1(1) specifies that AES be used to perform encryption and decryption operations for the various protocols specified in the PP. FCS_COP.1(2) requires a digital signature capability be implemented in the TOE for trusted updates and certificate operations associated with identification and authentication of authorized IT entities and remote administrators. FCS_COP.1(3) and FCS_COP.1(4) require that the TSF provide hashing services using an implementation of the Secure Hash Algorithm algorithms for data integrity verification and non-data integrity

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
		operations. FCS_RBG_EXT.1 ensures that keying material is robustly generated. FIA_X509_EXT.1 requires that the certificates used to support many of the cryptographic operations previously mentioned conform to an appropriate standard.
O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	FTA_TAB.1	FTA_TAB.1 requires the TOE to display an administrator defined banner before a user can establish an authenticated session. This banner is under complete control of Authorized Administrators in which they specify any warnings regarding unauthorized use of the TOE.
O.FAIL_SECURE The TOE shall fail in a secure manner following failure of the power-on self tests.	FPT_FLS.1	FPT_FLS.1 requires that on a detected failure the TOE maintains a secure state.
O.PROTECTED_COMMUNICATIONS The TSF shall protect TSF data when it is in transit between the TSF and another trusted IT entity.	FAU_STG_EXT.1 FCS_IPSEC_EXT.1 FCS_TLS_EXT.1 FCS_SSH_EXT.1 FCS_HTTPS_EXT.1 FTP_ITC.1 FTP_TRP.1 FIA_8021X_EXT.1 FPT_RPL.1	FAU_STG_EXT.1 protects the audit records through transmission between external audit storage. FTP_ITC.1 and FTP_TRP.1 (and the supporting protocols 802.11-2007, FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1, and FCS_HTTPS_EXT.1) require the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification. FIA_8021X_EXT.1 provides the two-way authentication necessary to allow a wireless client access to the wired network, and serves as a part of the 802.11-2007 WPA2 protocol to establish the communication channel with the wireless client. FPT_RPL.1 ensures that administrator sessions or data communicated with an authorized IT entity cannot be replayed. *Application Note: The ST author will modify the rationale to reflect the protocols that are implemented by the TOE.*
O.PROTOCOLS The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability.	FCS_IPSEC_EXT.1 FCS_TLS_EXT.1 FCS_SSH_EXT.1 FCS_HTTPS_EXT.1 FTP_ITC.1 FIA_8021X_EXT.1	FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1, FCS_HTTPS_EXT.1, FTP_ITC.1 (for 802.11-2007) and FIA_8021X_EXT.1 (in support of 802.11-2007) all reference the standards (and indicate any restrictions on those standards) applicable to the protocol they require to be implemented. *Application Note: The ST author will modify the rationale to reflect the protocols that are implemented by the TOE.*

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
O.REPLAY_DETECTION The TOE will provide a means to detect and reject the replay of authentication data and other TSF data and security attributes.	FPT_RPL.1	FPT_RPL.1 requires the TOE to detect and reject any attempted replay of authentication data from a remote user.
O.RESIDUAL_INFORMATION_CLEARING The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.	FCS_CKM_EXT.4 FDP_RIP.2	FCS_CKM_EXT.4 ensures the destruction of any cryptographic keys when no longer needed. FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to build network packets is either cleared or that some buffer management scheme be employed to prevent the contents of a packet being disclosed in a subsequent packet (e.g., if padding is used in the construction of a packet, it must not contain another user's data or TSF data).
O.RESOURCE_AVAILABILITY The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).	FRU_RSA.1	FRU_RSA.1 imposes quotas on exhaustible resources such that resources can be controlled and DoS attacks may be mitigated.
O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control an administrator's logical access to the TOE and to control administrative access from a wireless client.	FIA_AFL.1 FIA_PMG_EXT.1 FIA_UAU_EXT.5 FIA_UAU.6 FIA_UAU.7 FIA_UIA_EXT.1 FMT_SMR.1 FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4 FTA_TSE.1	FIA_AFL. provides a settable unsuccessful authentication attempt threshold that prevents unauthorized users acting remotely from gaining access to authorized administrator's account by guessing authentication data by locking the targeted account until the Authorized Administrator takes some action (e.g., re-enables the account) or for some Authorized Administrator defined time period. FIA_PMG_EXT.1 defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained. FIA_UAU_EXT.5 requires that the TSF provides local authentication methods (one of which is required to be a local password-based mechanism, with other optional (potentially off-box) mechanisms allowed) to ensure that unauthorized users cannot gain logical access to the TOE. FIA_UAU.6 requires a user to reauthenticate when a password is changed or the session is locked and FIA_UAU.7 ensures that authentication feedback is obscured at the local console. FIA_UIA_EXT.1 plays a role in satisfying this objective by ensuring that every user is identified and authenticated before the TOE performs any mediated functions. FMT_SMR.1 controls the administrator's ability to

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
		perform administrative actions from a wireless client; the capability must be disabled by default. FTA_SSL_EXT.1 provides the Authenticated Administrator the capability to specify a time interval of inactivity in which an unattended local administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources. FTA_SSL.3 takes into account remote sessions. After an Administrator-defined time interval of inactivity remote sessions will be terminated, this includes user proxy sessions and remote administrative sessions. This component is especially necessary since remote sessions are not typically afforded the same physical protections that local sessions are provided. FTA_SSL.4 provides administrators the capability to exit or logoff administrative sessions, rather than wait for the session to be terminated. FTA_TSE.1 contributes to this objective by limiting a user's ability to logically access the TOE. This requirement provides the ability to deny remote administrators access to the TOE based on time and day(s) of the week and location (e.g., from a specific port number, IP address, etc).
O.SESSION_LOCK The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.	FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4	FTA_SSL_EXT.1 provides an authenticated Administrator the capability to specify a time interval of inactivity in which an unattended local administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources. FTA_SSL.3 takes into account remote sessions. After an Authorized Administrator defined time interval of inactivity remote sessions will be terminated, this includes user proxy sessions and remote administrative sessions. This component is especially necessary because remote sessions are not typically afforded the same physical protections that local sessions are provided. FTA_SSL.4 provides administrators the capability to exit or logoff administrative sessions, rather than wait for the session to be terminated.
O.SYSTEM_MONITORING The TOE will provide the capability to generate audit data and send those data to an external IT entity.	FAU_GEN.1 FAU_GEN.2 FAU_SEL.1 FAU_STG.1 FAU_STG_EXT.1 FAU_STG_EXT.3	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. FAU_GEN.2 ensures the audit records associate a user identity with the auditable event. FAU_SEL.1 allows the administrator to configure which auditable events will be recorded in the audit trail.

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
	FPT_STM.1 FAU_SAR.1 FAU_STG_EXT.4	FAU_STG.1 requires some amount of local audit storage which must be protected from unauthorized access. FAU_STG_EXT.1 protects the audit records through transmission between external audit storage. FAU_STG_EXT.3 defines the set of events that must occur when the link to the external audit storage is not available. FPT_STM.1 requires that the TOE be able to provide reliable time stamps for use in audit records. FAU_SAR.1 Allows administrators the ability to read and interpret audit records to aid in system monitoring. FAU_STG_EXT.4 allows an authorized administrator decide how to prevent the loss of audit data to keep the desired audit information.
O.TIME_STAMPS The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps	FPT_STM.1	FPT_STM.1 requires that the TOE be able to provide reliable time stamps for its own use and therefore, partially satisfies this objective. Time stamps include date and time and are reliable in that they are always available to the TOE, and the clock must be monotonically increasing.
O.TOE_ADMINISTRATION The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.	FIA_PMG_EXT.1 FIA_UAU_EXT.5[19] FMT_MTD.1(1)-(3) FMT_MOF.1 FMT_SMF.1 FMT_SMR.1 FTP_TRP.1 FAU_SAR.2	FIA_PMG_EXT.1 defines management capabilities and requirements for administrator specification of password/secret strength. FIA_UAU_EXT.5 requires that the TSF provides local authentication methods (one of which is required to be a local password-based mechanism, with other optional (potentially off-box) mechanisms allowed) to ensure that unauthorized users cannot gain logical access to the TOE. FMT_MTD.1 and FMT_MOF.1 restrict the ability to manage certain functionality and identify security attributes of an authorized administrator. FMT_SMF.1 specifies the management functions that an only administrator must perform. FMT_SMR.1 defines at least one administrator role (Authorized Administrator) to perform administrative actions. The TSF is able to associate a human user to this role. FTP_TRP.1 requires that the TSF provide a trusted path for remote administration. FAU_SAR.2 prevents access to audit records in the audit trail except to authorized administrators.

[19] This requirement was written as FIA_UAU.5 in the PP. The ST author assumed this was a typo and updated this to the applicable SFR.

Table 10: TOE Security Functional Requirements Rationale		
Objective	SFR Addressing the Objective	Rationale
O.TSF_SELF_TEST The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.	FPT_FLS.1 FPT_TST_EXT.1	FPT_FLS.1 requires that on a detected failure the TOE maintains a secure state. FPT_TST_EXT.1 requires the TOE to provide a suite of self tests to assure the correct operation of the TSF.
O.VERIFIABLE_UPDATES The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.	FCS_COP.1(2) FCS_COP.1.(3) FPT_TUD_EXT.1	FCS_COP.1(2) and FCS_COP.1(3) specify digital signature algorithms and hash functions used in verification of updates. FPT_TUD_EXT.1 provides a way to determine the version of firmware running, initiate an update, and verify the firmware/software updates to the TOE prior to installation.
O.WIRELESS_CLIENT_ACCESS The TOE will provide the capability to restrict a wireless client in connecting to the TOE.	FTA_TSE.1	FTA_TSE.1 provides the capability to control access by wireless clients based on time of day, their location (e.g., IP address), and other attributes that may be implemented by the TOE.

7. TOE Summary Specification

7.1 Implementation description of TOE SFRs

This section provides evaluators and potential consumers of the TOE with a high-level description of how each SFR is implemented, thereby enabling them to gain a general understanding of the evaluated functionality. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security requirements.

7.2 TOE Security Functions

The TOE consists of the following Security Functions:

- Security Audit
- Cryptography
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- Resource Utilization
- TOE Access/Trusted Path

7.3 Security Audit

The TOE generates an audit record of the following events in addition to those items specified in FAU_GEN.1 (a-c):

Table 11: Audit Record Events

Requirement of Interest	Auditable Events	Additional Audit information
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	None.
FAU_STG_EXT.3	Loss of connectivity.	None.
FCS_CKM.1(1)	None.[20]	None.
FCS_CKM.1(2)	None.[21]	None.
FCS_CKM.2(1)	Failure of the key distribution activity.	None.
FCS_CKM.2(2)	Failure of the key distribution activity, including failures related to wrapping the GTK.	Identifier(s) for intended recipients of wrapped key.
FCS_CKM_EXT.4	Failure of the key zeroization process.	None.[22]
FCS_COP.1(1)	Failure of encryption or decryption.	None.[23]
FCS_COP.1(2)	Failure of cryptographic signature.	None.[24]

[20] Updated per TD0036

[21] Updated per TD0036

[22] Updated per TD0036

[23] Updated per TD0036

FCS_COP.1(3)	Failure of hashing function.	None.[25]
FCS_COP.1(4)	Failure in Cryptographic Hashing for Non-Data Integrity.	None.[26]
FCS_COP.1(5)	Failure of WPA2 encryption or decryption.	None.[27]
FCS_IPSEC_EXT.1	Protocol failures. Establishment/Termination of an IPsec SA. Negotiation "down" from an IKEv2 to IKEv1 exchange.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_RBG_EXT.1	None.[28]	None.
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken (e.g., disabling of an account) and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal).	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.5	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.6	Attempts to re-authenticate.	Origin of the attempt (e.g., IP address).
FIA_8021X_EXT.1	Attempts to access to the 802.1X controlled port.	Provided client identity (IP address).
FIA_X509_EXT.1	Attempts to load certificates. Attempts to revoke certificates.	None.
FPT_FLS.1	Failure of the TSF.	Indication that the TSF has failed with the type of failure that occurred.
FPT_TST_EXT.1	Execution of this set of TSF self-tests. Detected integrity violations.	For integrity violations, the TSF code file that caused the integrity violation.
FPT_TUD_EXT.1	Initiation of the update. Any failure to verify the integrity of the update.	No additional information.
FRU_RSA.1	Maximum quota being exceeded.	Resource identifier.
FTA_SSL_EXT.1	Locking of an interactive session by the session locking mechanism. Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	Terminating a session by quitting or logging off.	None.
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism.	Reason for denial, origin of establishment attempt.
FTP_ITC.1	All attempts to establish a trusted channel. Detection of modification of channel data.	Identification of the initiator and target of channel.

[24] Updated per TD0036

[25] Updated per TD0036

[26] Updated per TD0036

[27] Updated per TD0036

[28] Updated per TD0036

FTP_TRP.1	All attempts to establish a remote administrative session. Detection of modification of session data.	Identification of the initiating IT entity (e.g., IP address).
FCS_TLS_EXT.1	Protocol failures. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_SSH_EXT.1	Protocol failures. Establishment/Termination of an SSH session .	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_HTTPS_EXT.1	Protocol failures. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.

The TOE supports remote audit logging using the syslog standard with an external server. The TOE allows the user to filter audit logs via administrator identity, event type, and user interface.

Audit messages are entered into the log and the subset of the log contents are sent to the syslog server according to the filters as opposed to limiting which messages are entered into the log according to the filter criteria.

When an administrative command is executed, the TOE sets up the session data structure which includes the “user identity”. When an audit log is generated, the session data is passed along with the audit information and the TOE simply extracts the “user identity” from the session data structure.

The TOE generates one or more of the following audit log messages in the local log during startup (or when a user requests a reboot):

- SUCCESS Modifying welcome banner
- FIPS Power-up self-tests completed successfully
- Rebooting controller now

To send audit log messages from the TOE to an external server, the function must be enabled and the TOE’s connection to the external syslog server must be configured and enabled. Logged events of every severity level can be sent to the remote server, or the TOE can be globally configured to send only a subset of messages, filtered by severity level, for audit logging.

Additionally, the filtering of administrative event logs by User Interface (MAC address), Fortress Security and Interface type (as described by Section 7.3.1 and 7.3.2) apply only when the administrator is logged on from a MAC address that is not itself subject to the separately configured MAC Auditing Settings. If an administrator logs on, and the source MAC address is from a listed MAC address, the audit logging configuration for that MAC address is applied

In Advanced View, after the TOE’s internal clock has been set to within 1000 seconds of the current time on the network, the TOE can be enabled to synchronize its clock with the time disseminated by up to three configured NTP servers. Once the TOE’s system clock is successfully synchronized with NTP server time, NTP manages the drift between the time on the TOE (the NTP client) and the time maintained by the NTP server(s) for the network. If the TOE is out of sync with NTP server time, the NTP daemon automatically corrects the TOE’s system clock. The TSF uses its system clock for audit timestamps.

The way in which administrative activity on the TOE is filtered can be globally configured for audit logging. Global settings apply to an administrative session only when the Audit setting for the administrator's individual account is set to "Auto". At the default Audit setting of Required, all activity on an administrative account is sent to the audit log without regard to global settings. For all audit actions associated with an administrative user, the audit log includes that user name. **FAU_GEN.1 FAU_GEN.2**

The TOE keeps 3.5 Mbytes of local audit log data in a 20 Mbyte partition. There are no users that can access this partition. The partition cannot be deleted since the user has no access to the shell. Access to the shell is necessary to issue a command to delete or format the partition. Within this space is the current log file and the two most recent log files that have been rotated. These log files are rotated as they fill up. The process for log rotation is as follows: log files are filled by audit event logs as they are generated. When that log file is full (i.e. there is no room for additional logs) a new log file is used to place audit event logs in. Since there are only three log files in rotation, the TSF overwrites the oldest audit log file upon audit log rotation when all three audit log files are currently full. When the TSF sends audit log data to the external syslog server, all data is encrypted with an IPsec tunnel. The log messages are sent when they are generated. The TOE uses Syslogd 1.5.0 compatible with RFC 3164. The granularity of the timestamps is 1 second. It is possible that multiple audit messages are logged within the granularity of the time stamps (1 second). The syslog design utilizes socket(s) to stream the audit log messages to syslogd. The syslogd process sends out UDP packets tunneled within the IPSec TCP tunnel which guarantees order of transmission Therefore, messages are sent in the order they are generated, If there is no link or the link goes down to the audit server, the TSF adds a "Communication error" to the local log. **FAU_STG_EXT.1, FAU_STG.1, FAU_STG_EXT.3, FAU_STG_EXT.4**

All administrative accounts can view logs. One administrative role is a Log Viewer level account. If the user logs on to a Log Viewer-level account in the GUI, the GUI opens on the System Log screen. Administrator- and Maintenance-level administrators can view the entire log, while Log Viewer-level administrators can view only nonconfiguration events.

The TOE's three status icons indicate the severity of System Log messages:

- Notice or Info - message is purely informational
- Warning - unexpected event may indicate a problem/require attention
- Error - failure or attempted breach requires attention

The controls at the lower right of the screen can be used to page through the log and specify the number of messages shown per page: 10, 20, 40 or 60. **FAU_SAR.1, FAU_SAR.2**

7.3.1 User Interface and Fortress Security Status

Administrative activity sent to the audit log can be filtered by the kind of management interface the administrator is logged on through and whether the interface is encrypted or clear, wired or wireless:

- Audit by User Interface - There are three ways an administrator can access the TOE:
  - Console - a serial connection to the chassis Console port
  - SSHv2 - a Secure Shell connection to the TOE CLI
  - GUI - an HTTPS (Hypertext Transfer Protocol Secure) connection to the TOE GUI
- Audit by Fortress Security – This specifies generating audit logs on only an encrypted, or only a clear interface. All remote management connections to the TOE must be made on one of its Clear Interfaces (on which Fortress Security is Disabled) or on one of its Encrypted Interfaces (on which Fortress Security is Enabled).

- Audit by Interface Type - All remote management connections must be made through either a Wired interface (Ethernet port) or a Wireless interface, a BSS (Basic Service Set is an access point associated with one or more stations) on one of the TOE's radios.

The TOE handles audit event logging according to a hierarchy of categories, ordered as shown above (Audit by User Interface, Audit by Fortress Security, Audit by Interface Type). Each of the interface and Fortress security status controls for audit event logging can be set to one of three behaviors:

- Required - events originating from that interface or from an interface with the specified Fortress security status are logged, provided they are not prohibited in a superior audit setting.
- Prohibited - events originating from that interface or from an interface with the specified Fortress security status are not logged, provided they are not Required in a superior audit setting
- Auto - events originating from that interface or from an interface with the specified Fortress security status are logged according to whether they are Prohibited or Required in a superior setting. If all applicable superior settings are at Auto, events are logged according to any applicable inferior settings.

Events are checked against the audit settings for User Interface, Fortress Security and Interface Type, in that order, and logged according to the first applicable "Required" or "Prohibited" (as defined in the first two bullet points in this section) setting. The TOE logs all authentication attempts (successes and failures), as well as all failures to encrypt or decrypt data, and all changes to the TOE security configuration. These security events are always logged, regardless of configuration. This allows the user to select auditing based on success or failure of security events. Audit logging is Required by default for all interfaces, regardless of user, type, or Fortress security status. In addition, events can be audited based upon whether the audit message is associated with an encrypted entity or not. If an event is associated with an encrypted entity, the event is part of the encrypted zone. Conversely, if an event is associated with a non-encrypted entity it is part of the clear zone.

7.3.2 Logging Administrator Activity by Event Type

The events can be sent to the audit log can be specified by three broad types:

- Login - When Enabled, logon activity by subject administrators can be sent to the audit log. When Login is Disabled, the logon activity of subject administrators are not sent.
- Security - When Enabled, if Configuration (below) is also Enabled, any changes made by subject administrators to the TOE's security settings can be sent to the audit log. When Security is Disabled, security reconfiguration by subject administrators are not sent.
- Configuration - When Enabled, if Security (above) is also Enabled, all changes made the administrators to the TOE's configuration can be sent to the audit log. If Security is disabled when Configuration is Enabled, all changes except those to security settings can be logged. When Configuration is Disabled, TOE reconfigurations by subject administrators are not sent (even if Security logging is Enabled).

In addition to the conditions described in this section (7.3), whether or not events of an Enabled type are actually sent to the audit log depends on whether the event meets the interface and Fortress security status criteria for audit logging configured in the rest of the Global Auditing Settings frame (below). All three event types are Enabled by default. **FAU_SEL.1**

7.4 Cryptography

The TSF also employs and supports standards and protocols-based network security measures, including: RADIUS (Remote Authentication Dial in User Service), WPA2 (Wi-Fi Protected Access), and IPsec (Internet Protocol Security). The TSF can be configured to operate using FIPS CAVP certificates. (Appendix B: FIPS Compliance**.,** Table 15: CAVP Certificate Reference).

7.4.1 Cryptographic Key Management

For cryptographic key distribution of the Pairwise Master Key, the PMK is transferred through the MS_MPPE_SEND_KEY Vendor Specific Attribute (VSA). **FCS_CKM.2(1)**

For cryptographic key distribution of the Group Temporal Key (GTK), the GTK is wrapped using the AES Key Wrap algorithm specified in RFC 3394. The key used is the KEK derived in the 802.11i four-way handshake performed when each client connects to the TOE. **FCS_CKM.2(2)**

For cryptographic key generation of asymmetric keys, the TOE conforms to

- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes. The TOE conforms to NIST SP800-56A 6.1.2.1 dhEphem, C(2, 0, FFC DH) and NIST SP800-56A 6.1.2.2 dhEphem, C(2, 0, ECC CDH).
- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes.
- NIST Special Publication 800-56B, "Recommendations for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptograpy" for RSA-based key establishment schemes.

The TOE conforms to all shall, should, and should not statements in these sections. There are no must, must not, or shall not statements in the listed section. **FCS_CKM.1(2)**

All cryptographic primitives are defined and implemented consistently as specified by the FIPS accompanying algorithms certs. Refer to Appendix B: FIPS Compliance for specific FIPS information. The TOE has also been certified by the Wi-Fi alliance conforming to their well-publicized standards for interoperability and cryptographic standards. **FCS_CKM.1(1)**

The configuration database is stored in a file that has been hashed using SHA160. It is then encrypted using cipher block chaining. All encrypted keys which are decrypted have their memory usage zeroized after the usage is completed by writing all 0's. The following is a list of the secret keys (keys used for symmetric encryption), private keys, and critical security parameters used to generate keys, all of which are stored in the configuration database in a flash file system:

- Administrative passwords
- WPA2 keys
- Authentication server keys
- Device Access ID
- Public/private key pairs
- X.509 certificates
- IPsec pre shared keys

FCS_CKM_EXT.4

7.4.2 Cryptographic Operation

The TOE performs AES encryption by means of FIPS Approved AES algorithms. The TOE performs AES encryption and decryption as specified by FIPS PUB 197, "Advanced Encryption Standard (AES)", and NIST SP 800-38A. The TSF implements the following modes and key sizes:

- Modes
  - AES-CBC
- Key Sizes
  - 128 bits
  - 192 bits
  - 256 bits

FCS_COP.1(1)

The TSF performs cryptographic signature services in accordance with the RSA Digital Signature Algorithm (rDSA) with a key size of 2048 bits. **FCS_COP.1(2)**

The TOE performs hashing using a software library, which meet FIPS Pub 180-2, "Secure Hash Standard." approved hash algorithms. A block of data and a salt value are passed in, and a digest and its length is returned. The following hash ciphers are used:

- Algorithm
  - SHA-1
  - SHA-256
  - SHA-384
- Message Digest Sizes
  - 160 bits
  - 265 bits
  - 384 bits

FCS_COP.1(3)

The TOE performs keyed-hash message authentication using FIPS Approved algorithms which meets FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-3, "Secure Hash Standard**"** A block of data, a block length, and a key value are passed in, and a digest and its length are returned. The TOS implements the following HMAC ciphers:

- HMAC algorithms
  - SHA-1
  - SHA-256
  - SHA-384
- Key Sizes
  - 160 bits
  - 256 bits
  - 384 bits

FCS_COP.1(4)

The TOE performs WPA2 encryption/decryption on wireless traffic by having the radio driver use FIPS Approved algorithms and meets FIPS PUB 197, NIST SP 800-38C, and IEEE 802.11-2007. A block of data, a key, and a block mode are passed in, and an encrypted/decrypted block and size are returned. The encryption and decryption is performed by the AES CCMP algorithm with a key size of 128 bits.

7.4.3 Zeroization

The configuration database is stored in a file that has been hashed using SHA160. It is then encrypted using cipher block chaining. The key used to encrypt the configuration database is stored in I2C (meaning, it is set onto the EPROM when the box is manufactured). The key on the EPROM is never zeroized, since without it the box is not operational. This key is never used for communication. All encrypted keys which are decrypted have their memory usage zeroized after the usage is completed by writing all 0's. The following is a list of the secret keys (keys used for symmetric encryption), private keys, and critical security parameters used to generate keys, all of which are stored in the configuration database in a flash file system:

- Administrative passwords
- WPA2 keys
- Authentication server keys
- Device Access ID
- Public/private key pairs
- X.509 certificates
- IPsec pre shared keys

FCS_CKM_EXT.4

7.4.4 Cryptographic Protocols

The TOE uses IPsec to secure communications to the RADIUS server, the Syslog server, and the NTP server. When establishing a tunnel, the TOE only operates in tunnel mode and the TOE ensures that the "confidentiality only" ESP security service is disabled when presented with an IKE proposal for ESP with no integrity. As an IKE initiator, the TOE ignores Security Association payloads containing an ESP "confidentiality only" proposal. The lifetimes for IKEv1 SAs (both Phase1 and Phase 2) are established by being fully configurable at the time the cryptography parameters are defined. These lifetimes may be configured for number of seconds and/or bytes sent. The TOE does not use aggressive mode for IKE v1. For the IKE peer authentication process, the TOE performs IKEv1 consistently with Section 1.5 of RFC 2408, and 2407. The TOE performs IKEv2 consistently with Section 2.15 of RFC 4306. When the TOE is performing an IKE Diffie-Hellman key exchange the secret value "x" is 224/256/384 bits generated by NIST SP800-90 HMAC DRBG, as specified by FCS_RBG_EXT.1for DH groups 14/19/20 respectively. The probability that any nonce is repeated during the life of a specific SA is less than 1 in 2^256, which is sufficient for any negotiated cipher suite. The DH groups implemented and used by the TOE are DH Groups 14 (2048-bit MODP) and 19 (256-bit Random ECP), 20 (384-bit Random ECP). For IPSec, the determination of the DH group is made by CLI commands. Pre-shared keys are used in authentication of IPsec connections in version 1 of the Internet Key Exchange (IKE) protocol as documented in Section 1.5 of RFC 2408. Pre-shared keys are used in authentication of IPsec connections in version 2 of the Internet Key Exchange (IKE) protocol as documented in Section 2.15 of RFC 4306. Pre-shared keys are established by the administrator using either the GUI or CLI interfaces. Pre-shared keys may be specified as strings of ASCII characters or as a sequence of hexadecimal digits. IPsec keys must be between 16 and 128 ASCII characters, or between 32 and 256 hex digits in length. Pre-shared keys may also be generated randomly using a NIST SP800-90 compliant DRBG. IPsec uses the following encryption ciphers:

- AES128
- AES256

Operating in CBC mode.

The following is a list of algorithms that are allowed for IKE and ESP exchanges and their bits of security.

Table 12: Allowed Algorithms for IKE and ESP Exchanges			
StrongSwan	Algorithm	DH Group	Bits of Security for DH Group
IKE			
aes128-sha1-modp2048!	AES-CBC-128	14	112
aes256-sha1-modp2048!	AES-CBC-256	14	112
aes128-sha256-ecp256!	AES-CBC-128	19	128
aes256-sha384-ecp384!	AES-CBC-256	20	192
ESP			
aes128-sha1-modp2048!	AES-CBC-128	14	112
aes256-sha1-modp2048!	AES-CBC-256	14	112

FCS_IPSEC_EXT.1

When establishing an SSH tunnel, the TOE allows the following ciphers:

- Public key algorithms
  - SSH_RSA
- Encryption algorithms
  - AES-CBC-128
  - AES-CBC-256
- Data integrity algorithms
  - HMAC-SHA1
  - HMAC-SHA1-96
- Key exchange
  - diffie-hellman-group14-SHA1

An administrative user can authenticate with SSH public key authentication and a user name and password or with just a user name and password. If that user has established a session, then that user is given a 60 second timeout window before that session expires. For SSH, the timeout counter is reset when there is keyboard activity. The GUI also has a 60 second timeout counter and is reset when the user interfaces with the GUI (such as pressing a button and submitting login credentials) If a user enters three failed authentication attempts in a single session, then the TOE locks out that administrative user's account. If that user enters more than three failed authentication attempts across multiple sessions within an hour then the TOE also locks that user's account. The TOE implements the SSH protocol using OpenSSH v5.8 P1. This industry standard implementation monitors incoming packet size by counting the number of bytes. If the byte threshold exceeds 32768, then the TSF drops that packet. The TOE also limits the amount of traffic that can pass in an SSH tunnel before requiring to be re-negotiated. This is set at 2 Gigabytes. This is effectively more restrictive than 2^28 packets. For SSH, the DH setting is determined based upon the offer made by the client and the local configuration setting on the TOE.
FCS_SSH_EXT.1

The TSF provides testing which consists of the minimum entropy test from NIST SP800-90, appendix C. The lowest allowed min-entropy is 80% or 4.8 bits entropy per 6-bit sample. Anything less than that and the FIPS test fails and places the device into a failed state. The continuity test catches repeat values. The

TSF tests the actual "randomness" by doing a min-entropy test. The RBG is always seeded with a minimum of 256 bits of entropy. **FCS_RBG_EXT.1**

The TOE uses the TLS 1.0 protocol for securing communication with the GUI through HTTPS/TLS, as well as adding additional security in communicating with the RADIUS authentication server. The TOE provides TLS for the Web Server(https) services. The authentication server provides EAP-TLS for authentication for WPA2 via x.509 certificates. The TLS implementation allows the following cipher suites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

FCS_TLS_EXT.1.1

The TOE uses HTTPS, which is defined as HTTP over SSL, which in turn uses TLS. The TOE requests the client for a certificate. Login credentials are required at log in page and pass through the established TLS connection. **FCS_HTTPS_EXT.1**

7.5 User Data Protection

When the TOE is constructing a PDU (protocol data unit), it makes any previous information unavailable when it is allocated for the next PDU. The PDU is not padded at all as a part of normal packet processing. Data passing into the system is copied from the driver that initially received that data into a PDU buffer of exactly the right size. There is no need to pad or zeroize data since the buffer is the correct size and there is nothing to pad/zeroize.

For IPsec:

- Only IPsec-tunnel mode is supported, so the original IP header is encrypted.
- The decrypted IPHDR.length must be <= the encrypted IPHDR.length
- The frames are protected with a MIC.

In general:

- When the network driver allocates a PDU buffer, 2 FP (fast path) working buffers are allocated, one for the incoming PDU and one for the resulting PDU (encrypt/decrypt).
- The FP working buffers are larger than the supported MTU + encrypt/decrypt overhead.
- The buffer processing within the FP is protected by a wrapper object. This wrapper will enforce the buffer boundaries.
- The crypto device will also abort the FP buffer if its length exceeds those boundaries.
- After the crypto device processes (encrypt/decrypt) the frames, the network driver will transmit based upon the result length, not the allocated buffer size.

The data from the previous PDU is, therefore only made unavailable when that specific part of memory is allocated to the next PDU and overwritten with new data. **FDP_RIP.2**

7.6 Identification and Authentication

The behavior of the TOE when encountering unsuccessful authentication attempts is configurable. The TOE always logs authentication attempts. The configuration options available are to lock the user out until an administrator unlocks them, or locking them out for a specified amount of time after N unsuccessful attempts. The number of unsuccessful attempts, the lockout duration, and lockout until

explicitly unlocked by an administrator are all configurable. In addition, the TOE fully logs unsuccessful attempts as well as the interface the attempt came in on. The TOE tracks the unsuccessful authentication attempts for account locking by the user name. If the user is locked out, the TOE does not even accept the correct username/password authentication entry. **FIA_AFL.1, FIA_UIA_EXT.1**

A successful authentication is determined by either a successful username and password combination, or additionally required public key/certificate for SSH/TLS respectively. A failure to find a public key and/or incorrect password will result in a failed authentication attempt. When a user is entering their password information, the password is obscured such that no observer could read the password off the screen. This is done by using a circle to represent all characters while accessing the local (console via RS-232) administrative interface. The admission can be handled by either a local authenticator or a RADIUS server. In the local case, passwords entered are converted into a SHA-256 digest using a salt value. This is compared to the digest value for that user. No passwords are ever stored as clear text. For remote authentication the TOE must have a connection to the RADIUS server. Communications to the RADIUS server are secured using an IPsec tunnel and the TLS protocol.

An administrative user is required to re-authenticate when that user changes their own password, and following a TSF-initiated locking as described in any of the FTA_SSL requirements in this ST. There are two TSF responses allowed prior to administrative authentication. The TSF displays the access banner warning and sends and receives MVP (Mesh Viewer Protocol) packets. Every 30 seconds the TSF sends out MVP packets to all other Fortress nodes. These packets include information on the TOE (IP address, MAC, type (i.e. ES810, ES2440, etc.) and location (manual or obtained by GPS if enabled). It also contains for each link that the box has, the MAC and IP of the other endpoint of the link, as well as the signal strength of the link at the time the packet was created. While this information is available prior to authentication, these responses are only available via the trusted IPSEC channel, requiring appropriate X.509 certificate or pre-shared keys.**FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU.6, FIA_UAU.7**

A user can use X.509 certificates for TLS and IPsec. Certificates are stored in the configuration database. Access to the configuration database is from software only (meaning there is no means for a user to access it). The configuration database is encrypted and is not viewable. Certificates may be displayed ONLY to administrative users via the CLI or the GUI. **FIA_X509_EXT.1**

The TOE implements 802.1x-2010 by using hostapd, version 0.7.3. The product development process tests that the TOE conforms to the RFCs. These methods include peer code reviews, unit tests, nightly system automation tests, and formal QA testing. All sections of 802.1x-2010 features are supported. **FIA_8021X_EXT.1**

The TOE uses pre-shared keys for IPsec and WPA2. The TOE supports WPA2_PSK lengths of 8 to 63 ASCII characters or 64 hex digits. IPsec PSK keys must be between 16 and 128 ASCII characters, or between 32 and 256 hex digits in length. They must be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”). The TOE conditions the text-based pre-shared keys using the SHA-256 hash algorithm and can accept and generate bit based pre-shared keys using the random bit generator as specified in FCS_RBG_EXT.1.**FIA_PSK_EXT.1**

7.7 Security Management

For users that are not administrative users (wireless clients) there are no TSF commands or TSF data that is available to that user except the pre login access banner. Once a wireless client successfully authenticates with WPA2-PSK or EAP-TLS, that user can only elicit data through the TOE using the general WLAN functionality. This prevents any unprivileged configuration of the TOE or viewing of TSF data. **FMT_MOF.1, FMT_MTD.1(1)**

All passwords are stored as a hashed SHA-256 digest. A salt value used in conjunction with the digest cannot be seen by the user. When a user enters their password, a hashed SHA-256 digest is created with the known salt value. The newly created digest is then compared with the stored digest to determine if the login is successful. Furthermore, the entire configuration database is then encrypted using cipher block chaining (AES256-CBC) with a master key. There are no clear-text keys stored that must be zeroized. None of the key material used is visible in any way to the user, since there are no interfaces that allow the viewing of the Master Key. **FMT_MTD.1(2), FMT_MTD.1(3)**

The two remote administrative interfaces are the GUI and the SSH console interfaces. These allow the administrator to perform all security functionality as required by this PP and specifically FMT_SMF.1. Through the administrative interfaces, the following roles of: administrator, maintenance, and log viewer can access their allowed privileges and are maintained by the TSF. **FMT_SMF.1, FMT_SMR.1**

7.8 Protection of the TSF

The TOE runs a suite of self-tests on boot up. The following is a list of self-tests performed by the TOE:

- RAM Test: Performs a brief memory test of all RAM not used by boot loader and stack (where parameters and local variables are allocated from). The RAM test iterates over the physical RAM of the device, setting a series of fixed values and reading them back to ensure the memory was written and read properly each time. The last set of values written are all zeroes to ensure the memory is started from a zeroed state.
- Flash Test: Verifies the checksum of the entire Boot flash. The Flash test reads every byte of the flash image and uses those values to calculate a modular checksum over the image and compares the computed checksum to the stored checksum.
- Firmware Integrity Test: Verifies the integrity of the firmware by verifying the digital signature using rDSA with a key size of 2048.
- EEPROM Test: Verifies that the EEPROM can be written to and read from. The EEPROM Test reads and writes a small number of bytes to the EEEPROM device with known values at a test location within the EEPROM device and compares the result to ensure the EEPROM device can be read and written to.
- I2C Test: Probes each of the expected devices on the I2C bus to ensure the device responds to its address on the bus.
- [29]MDIO Test: Verifies that the [30]PHY [31]ID is as expected. The test performs a read of the [32]MII interface of each expected PHY address to ensure that the each expected Ethernet port is present and responds with the correct PHY Identifier, which consists of the correct Vendor ID and Device ID.

---

[29] Management Data Input/Output (MDIO), also known as Serial Management Interface (SMI) or Media Independent Interface Management (MIIM), is a serial bus defined for the Ethernet family of IEEE 802.3 standards for the Media Independent Interface, or MII. The MII connects Media Access Control (MAC) devices with Ethernet physical layer (PHY) circuits. The MAC device controlling the MDIO is called the Station Management Entity (SME).

[30] PHY refers to the physical layer of the OSI networking model.

[31] PHY ID is a physical layer register containing Vendor and Device ID. These values are simply byte values, which are set by for PHY chip for identification.

[32] MII was originally defined as a standard interface used to connect a Fast Ethernet (i.e., 100 Mbit/s) MAC-block to a PHY chip. A PHY chip refers to the physical layer of the OSI network model.

- PCI Test: Verify that the devices on the PCI bus are as expected by reading the device and vendor IDs. The PCI test utilizes a table of expected PCI devices, including the PCI bus address, PCI vendor ID, PCI device ID, PCI sub-vendor ID and PCI sub-device ID. The PCI bus is enumerated by listing every device on the bus and verifying that each expected device is at the correct bus address and each device is queried to ensure it has the correct PCI vendor ID, device ID, sub-vendor ID, and sub-device ID for that address.
- IDE Test: There are three parts to this test. It starts by reading from, writing to, and verifying the values in the IDE registers. It then executes the IDE device self-test and verifies the results. It then reads 100 random sectors.
- RTC Test: This test reads and saves current time. It then sets a known time/date that causes all dates/time to roll over and verifies that the rollover time is correct. It ends by restoring the current time.
- Watchdog Test: This test enables the watchdog timer. It then waits for the watchdog to time out and verifies that a timeout occurred.
- IRQ Test: This test starts by enabling CPU interrupts and then forcing the Ethernet PHY to cause an interrupt. It then verifies that the CPU received the interrupt.
- FPGA Test: This test checks the variations of available encrypt/decrypt (algorithm) engines. For each algorithm engine, the test sends known test data through that engine and verifies the results against known answers. It then generates 1000 packets randomly and performs a software based encrypt/decrypt on these packets using the system CPU (not the FPGA). These same packets are then sent through the engines and the results of the software based encrypt/decrypt are compared to the FPGA results.
- TPM Test: This tests RNG functionality. It does this by reading and extending the integrity registers, ensuring that the microcode has not been changed, and that the tamper-resistant and tamper-evident markers are under program control. The TPM also performs known answer tests for hashing, as well as for each symmetric and asymmetric algorithm it supports.

For key material and user data, the most critical security-related tests, such as the TPM test, the FPGA test, and any of the FIPS required tests, causes the box to stop operation as soon as the failure is detected. The FPGA is responsible for cryptographic operations as specified in Table 15: CAVP Certificate Reference. Since the FPGA is required to decrypt, the data is protected if the FPGA fails. Temporal keys are only stored as in use in working memory and any other keys material (such as passwords in the config file or shared secrets) are stored on the encrypted file system. Because of this, the TOE is always in a secure state. The failure of any critical security component causes the box to halt.

Once the TOE has completed the boot process, the entire suite of known answer tests and continuous tests are run. All tests must pass before the TOE begins handling user data or the administrator is able to log in. **FPT.FLS.1, FPT_TST_EXT.1.1**

For auditing, session establishment, SA (A Security Association is the establishment of shared security attributes between two network entities to support secure communication) lifetimes (the length of time until it SA is invalidated, a new key is generated, and the SA is re-negotiated) and X.509 certificate revocation, the internal clock is used. This is either set manually by the administrator, or by NTP. The connection to the NTP server is protected by an IPsec tunnel. **FPT_STM.1**

The TOE secures all communications with all IT entities using IPsec. It is through the IPsec protocol that the TOE detects replay attacks and rejects the data. IPsec ESP processing checks the sequence numbers and rejects packets if a duplicate is found. There is a CRC checksum computed at the PHY layer. If the checksum fails, the frames are dropped. If the frames are encrypted, an MIC is computed over the data. The MIC is verified during the decryption process, and if it fails, the data is dropped as well. **FPT_RPL.1**

Users can query the firmware/software version of the TOE and an authorized administrator can initiate updates to the TOE. When performing the update, the TOE compares the update files' signature using a certificate that comes pre-loaded on the device. As part of the build process, the update image is signed with a private key by GDMS. In this system, the "authorized source" is defined as the holder of the private key, thus making GDMS the only authorized source for updated images. Only if the signature is correct, the image can be installed. If an update is unsuccessful, a message is delivered to the user. Since the update process attempts to update a different partition than what is currently being run, the current active partition remains the same and the user continues to run the same code that was being run before the upgrade attempt was made. **FPT_TUD.1**

7.9 Resource Utilization

The TSF enforces a maximum number of simultaneous wireless connections to 127. This limits memory usage and number of virtual interfaces and buffers. This is imposed per subject (wireless connection) for simultaneous usage. Once the quota is reached, the TSF does not allow any additional wireless connections. **FRU_RSA.1**

7.10 TOE Access/Trusted Path

Once a user is identified and authenticated as an authorized administrator that user has access to their allowed TSF privileges and functions. An authorized administrator can specify a period of inactivity for the TSF to automatically terminate local and remote interactive sessions. An administrative user can initiate their own session termination prior to the specified inactive time period. **FTA_SSL_EXT.1.1, FTA_SSL.3.1, FTA_SSL.4.1**

Prior to an administrative user authenticating, that user is presented with an access display banner which displays an advisory notice and consent warning message regarding unauthorized use of the TOE. An authorized administrator can configure the TSF to deny establishment of a wireless client based on that client's location, time or day. The location is based on MAC address as the TOEs are rarely stationary in the intended environment. **FTA_TAB.1, FTA_TSE.1**

The TSF secures communications with all IT entities with IPsec. This includes RADIUS, syslog, and NTP. For RADIUS, TLS can be used in addition to IPsec. For TOE administration, the GUI, SSH(CLI) and local console CLI are available. The GUI and the remote CLI interfaces are secured using TLS/HTTPS and SSH respectively. The TLS is not included for all IT entities because they are already secured within the IPSec tunnel. TLS is not used to secure communications between RADIUS, syslog, and NTP is because customers will run those services within the trusted portion of the network. **FTP_ITC.1, FTP_TRP.1**

8 Appendix A: RFC Compliance

1. RFC 2246[33]
   The TLS Protocol Version 1.0, January 1999
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

2. RFC 2406
   IP Encapsulating Security Payload (ESP), November 1998
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

3. RFC2407
   The Internet IP Security Domain of Interpretation for ISAKMP, November 1998
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

4. RFC2408
   Internet Security Association and Key Management Protocol (ISAKMP), November 1998
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

5. RFC 2409
   The Internet Key Exchange (IKE), November 1998
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

6. RFC 2818
   HTTP Over TLS, May 2000
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

7. RFC 2865
   Remote Authentication Dial In User Service (RADIUS), June 2000
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

8. RFC 3164
   The BSD syslog Protocol, August 2001
   This document is on the list since it is referenced specifically in the WLAN protection profile. Specifically, it requires that all syslog messages conform to the specified severity levels.

   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

9. RFC 3394
   Advanced Encryption Standard (AES) Key Wrap Algorithm, September 2002
   Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

[33] The SFR FCS_TLS_EXT.1 incorrectly calls out 2346 when the related SFR for TLS is 2246.

10. RFC 3579
    RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP) September 2003
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

11. RFC 3602
    The AES-CBC Cipher Algorithm and Its Use with IPsec, September 2003
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

12. RFC 4109
    Algorithms for Internet Key Exchange version 1 (IKEv1), May 2005
    This RFC is unusual since all it does is updated the MUST/SHALL/SHOULD/MAY" status for the IKEv1 algorithms specified in RFC2409.
    Analysis: This table lists the algorithms updated in this RFC, the requirement in RFC 2409, the new requirement in RFC 4109, and the state of Fortress support for this requirement.

Table 13: RFC 4109 Analysis

Algorithm	RFC 2409	RFC 4109	Fortress Support
DES for encryption	MUST	MAY	No.
Triple DES for encryption	SHOULD	MUST	No
AES-128 for encryption	N/A	SHOULD	Yes
MD5 for hashing and HMAC	MUST	MAY (crypto weakness)	No
SHA-1 for hashing and HMAC	MUST	MUST	Yes
Tiger for hashing	SHOULD	MAY	No
AES-XCBC-MAC-96 for PRF	N/A	SHOULD	No
Pre-shared secrets	MUST	MUST	Yes
RSA with signatures	SHOULD	MAY	Yes
DSA with signatures	SHOULD	MAY	No
RSA with encryption	SHOULD	MAY	No
D-H Group 1 (768)	MUST	MAY	No
D-H Group 2 (1024)	SHOULD	MUST	Yes
D-H Group 14 (2048)	N/A	SHOULD	Yes
D-H elliptic curves	SHOULD	MAY	Yes

13. RFC 4251
    The Secure Shell (SSH) Protocol Architecture, January 2006
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

14. RFC 4252
    The Secure Shell (SSH) Authentication Protocol, January 2006Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

15. RFC 4253
    The Secure Shell (SSH) Transport Layer Protocol, January 2006

Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented. Please note that SSHv1 is supported but not configurable for use for security reasons.

16. RFC 4254
    The Secure Shell (SSH) Connection Protocol, January 2006
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented. Please note that SSHv1 is supported but not configurable for use for security reasons.

17. RFC 4301
    Security Architecture for the Internet Protocol, December 2005
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

18. RFC 4303
    IP Encapsulating Security Payload (ESP), December 2005
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

19. RFC 4306
    Internet Key Exchange (IKEv2) Protocol, December 2005
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

20. RFC 4307
    Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2), December 2005
    This RFC defines Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2).
    Analysis: This table lists the algorithms updated in this RFC, the requirement in sections 3.1.3, 3.1.4 and 3.1.5:

Table 14: RFC 4307 Analysis

Algorithm	RFC 4307	Fortress Support
D-H Group 2 1024 bit	MUST	Yes
D-H Group 14	SHOULD	Yes
ENCR-3DES	MUST-	No
ENCR_NULL	MAY	No
ENCR_AES_CBC	SHOULD+	Yes
ENCR_AES_CTR	SHOULD	No
PRF_HMAC_MD5	MAY	No
PRF_HMAC_SHA1	MUST	Yes
PRF_AES128_CBC	SHOULD+	Yes
DSA with signatures	MAY	No

21. RFC 4868
    Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec, May 2007
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

22. RFC 4945

The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX, August 2007.
Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

23. RFC 5216
    The EAP-TLS Authentication Protocol, March 2008
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

24. RFC 5280
    Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.
    Analysis: This TOE is restricted to functioning as a generator of CSR and a processor of the responses from the CA. The TOE does not act as the CA. All of the MUSTs, SHALLs, and REQUIRED statements for CSR generation and response processing are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

25. RFC 5430
    Suite B Profile for Transport Layer Security (TLS), March 2009
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

26. RFC 5996
    Internet Key Exchange Protocol Version 2 (IKEv2), September 2010
    Analysis: This RFC is fully supported in the product. All of the MUSTs, SHALLs, and REQUIRED statements are supported; all of the MUST NOT and SHALL NOT statements are not implemented.

9 Appendix B: Cryptographic Compliance

The following table contains FIPS algorithm certs:

Table 15: CAVP Certificate Reference

Algorithm	Cert #	Crypto Implementation	Library Version	Functionality	Operational Environment	Modes
AES	1520	Fortress Cryptographic Implementation - FPGA	2.0	IPsec (ESP) WPA2 (frame processing)	Xilinx Spartan FPGA	CBC (e/d; 128, 192, 256) CCM (KS: 128)
	3506	Fortress Cryptographic Implementation - SSL	2.1	IPsec (IKE) WPA2 (establishment) TLS SSH	AMD Alchemy MIPS Processor Broadcom XLS Processor	CBC (e/d; 128, 192, 256)
SHS	1358	Fortress Cryptographic Implementation - FPGA	2.0	WPA2 (frame processing) IPsec (ESP)	Xilinx Spartan FPGA	SHA-1 (BYTE-only) SHA-384 (BYTE-only)
	2891	Fortress Cryptographic Implementation - SSL	2.1	TLS SSH WPA2 (establishment) IPsec (IKE)	AMD Alchemy MIPS Processor Broadcom XLS Processor	SHA-1 (BYTE-only) SHA-224 (BYTE-only) SHA-256 (BYTE-only) SHA-384 (BYTE-only) SHA-512 (BYTE-only)
HMAC	890	Fortress Cryptographic Implementation - FPGA	2.0	WPA2 (frame processing) IPsec (ESP)	Xilinx Spartan FPGA	HMAC-SHA1 HMAC-SHA384
	2238	Fortress Cryptographic Implementation - SSL	2.1	TLS SSH WPA2 (establishment) IPsec (IKE)	AMD Alchemy MIPS Processor Broadcom XLS Processor	HMAC-SHA1 HMAC-SHA224 HMAC-SHA256 HMAC-SHA384 HMAC-SHA512

ECDSA	716	Fortress Cryptographic Implementation - SSL	2.1	IPsec WPA2 (establishment) TLS SSH	AMD Alchemy MIPS Processor Broadcom XLS Processor	FIPS186-4: SigVer: P-256: (SHA-1, 256) P-384: (SHA-1, 384)
	833					FIPS186-4: KeyGen: P-256, P-384
ECDSA Component-validation	573	Fortress Cryptographic Implementation - SSL	2.1	IPsec (IKE) WPA2 (establishment) TLS,	AMD Alchemy MIPS Processor Broadcom XLS Processor	ECDSA SigGen Component: P-256 & P-384
RSA	1800	Fortress Cryptographic Implementation - SSL	2.1	TLS SSH	AMD Alchemy MIPS Processor Broadcom XLS Processor	FIPS186-2: ALG[RSASSA-PKCS1_V1_5] SIG(ver): 2048, SHS: SHA-1
	1967					FIPS186-2: Key Gen: 2048 SIG(gen): 2048, SHA-256, SHA-384
DRBG 800-90	874	Fortress Cryptographic Implementation - SSL	2.1	TLS SSH WPA2 (establishment) IPsec (IKE)	AMD Alchemy MIPS Processor Broadcom XLS Processor	HMAC_Based DBRG: SHA-1, SHA-256, SHA-384, SHA-512
KAS	10	Fortress KAS Implementation	1.0	IPsec (IKE)	AMD Alchemy MIPS Processor Broadcom XLS Processor	FFC: SHA-256 ECC: P-256 SHA-256 HMAC ED: P-384 SHA-384 HMAC

DSA	1053	Fortress Cryptographic Implementation - SSL	2.1	TLS IPsec (IKE) SSH	AMD Alchemy MIPS Processor Broadcom XLS Processor	FIPS186-Key Gen: (2048, 224), (2048, 256), (3072, 256)

NOTE: The version included in the above table represents that of the crypto implementation and is common across all the TOE models included in this evaluation. The crypto implementation is versioned independently overall software image version since it can remain unchanged regardless of the other software components.

10 Acronyms

Table 16: TOE Related Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
CCMP	Counter Cipher Mode with Block Chaining Message Authentication Code Protocol
CLI	Command Line Interface
DA	Distributed Agent
EAP	Extensible Authentication Protocol
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESP	Encapsulating Security Payload
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
GTK	Group Temporal Key
IEEE	Institute of Electrical and Electronics Engineers
IKE	Internet Key Exchange
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
MDIO	Management Data Input/Output
MIC	Message Integrity Code
MII	Media Independent Interface
NATO	North Atlantic Treaty Organization
NSA	National Security Agency
NTP	Network Time Protocol

Table 16: TOE Related Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
PDU	Protocol Data Unit
PHY	The physical layer of the OSI model
PHY ID	A physical layer identifier
PSK	Pre-shared key
PMKSA	Pairwise Master Key Security Association
RAM	Random Access Memory
RTC	Real Time Clock
RADIUS	Remote Authentication Dial In User Service
RSN	Robust Security Network
SSH	Secure Shell
SNMP	Simple Network Management Protocol
TKIP	Temporal Key Integrity Protocol
TPM	Trusted Platform Module
UI	User Interface
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Table 17: CC Related Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
AES	Advanced Encryption Standard
AF	Authorization factor
AS	Authorization subsystem
CAVS	Cryptographic Algorithm Validation System
CC	Common Criteria
CCTL	Common Criteria Testing Laboratory
CM	Configuration management
COTS	Commercial Off-The-Shelf
CMVP	Cryptographic Module Validation Program
DRBG	Deterministic Random Bit Generator
DoD	Department of Defense
EAL	Evaluation Assurance Level
ES	Encryption Subsystem

Table 17: CC Related Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
FIPS	Federal Information Processing Standards
ISSE	Information System Security Engineers
IT	Information Technology
OSP	Organization Security Policy
PP	Protection Profile
PUB	Publication
RBG	Random Bit Generator
SAR	Security Assurance Requirements
SF	Security Function
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

11 References

Table 18: Supporting Documents

Reference	Description	Version	Date
[1]	Wireless Local Area Network (WLAN) Access System Protection Profile	1.0	December 1, 2011

Table 19: Common Criteria v3.1 References

Reference	Description	Version	Date
[2]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2009-07-001	V3.1 R3	July 2009
[3]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2009-07-002	V3.1 R3	July 2009
[4]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2009-07-003	V3.1 R3	July 2009
[5]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2009-07-004	V3.1 R3	July 2009

Table 20: TOE Guidance Documentation

Reference	Description	Version
[6]	Fortress Mesh Point and Network Encryptor Software GUI Guide	Version 5.4.5, Revision 1
[7]	Fortress Mesh Point and Network Encryptor Software CLI Guide	Version 5.4.5, Revision 2
[8]	Fortress Mesh Point Software Auto Configuration Guide	Version 5.4.5, Revision 1
[9]	Fortress ES520 Deployable Mesh Point Hardware Guide	Revision 3
[10]	Fortress ES820 Vehicle Mesh Point Hardware Guide	Revision 3
[11]	Fortress CC Operational GuidanceES Series v5.4.5.2240 Release CD	1.16