
# Aruba Remote Access Points and Aruba Mobility Controllers with ArubaOS 8.10

# Security Target

**Version 1.6**

**November 2023**

**Document prepared by**


www.lightshipsec.com

# Document History

| Version | Date | Author | Description |
|---|---|---|---|
| 1.0 | 17 August 2023 | L. TURNER | Release for certification. |
| 1.1 | 23 August 2023 | G. NICKEL | Updates to FCS_IPSEC_EXT.1 |
| 1.2 | 28 August 2023 | G. NICKEL | Update to TOE version, updates to FCS_SSHS_EXT.1 |
| 1.3 | 06 September 2023 | G. NICKEL | Address OR09 |
| 1.4 | 29 September 2023 | G. NICKEL | Updates to CAVP table, Guidance doc references, added TD's. |
| 1.5 | 05 October 2023 | G. NICKEL | Updates to Guidance Document references. |
| 1.6 | 08 November 2023 | G. NICKEL | Address ECR Comments |

# Table of Contents

# List of Tables

# 1 Introduction

## 1.1 Overview

1 This Security Target (ST) defines the Aruba Remote Access Points and Aruba Mobility Controllers with ArubaOS 8.10 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 The TOE is a distributed network security solution offered by Aruba, a Hewlett Packard Enterprise company and is comprised of Aruba Remote Access Points (RAP) and an Aruba Mobility Controller (each with an embedded ArubaOS).

## 1.2 Identification

**Table 1: Evaluation identifiers**

| | |
|---|---|
| **Target of Evaluation** | Aruba Remote Access Points and Aruba Mobility Controllers with ArubaOS 8.10<br>Evaluated Build: 8.10.0.8-FIPS |
| **Security Target** | Aruba Remote Access Points and Aruba Mobility Controllers with ArubaOS 8.10 Security Target, v1.6 |

## 1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) PP-Configuration for Network Device and Virtual Private Network (VPN) Gateways, Version 1.2, 31 March 2022 (CFG_NDcPP-VPNGW_V1.2)
This PP-Configuration includes the following components:

i) Base PP: collaborative Protection Profile for Network Devices, Version 2.2e (CPP_ND_V2.2E)

ii) PP-Module: PP-Module for Virtual Private Network (VPN) Gateways, Version 1.2 (MOD_VPNGW_V1.2)

e) NIAP Technical Decisions per Table 2

**Table 2: NIAP Technical Decisions**

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0527 | Updates to Certificate Revocation Testing (FIA_X509_EXT.1) | CPP_ND_V2.2E | Applicable |
| TD0528 | NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4 | CPP_ND_V2.2E | Applicable |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0536 | NIT Technical Decision for Update Verification Inconsistency | CPP_ND_V2.2E | Applicable |
| TD0537 | NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 | CPP_ND_V2.2E | Applicable |
| TD0546 | NIT Technical Decision for DTLS - clarification of Application Note 63 | CPP_ND_V2.2E | FCS_DTLSC_EXT.1 not claimed. |
| TD0547 | NIT Technical Decision for Clarification on developer disclosure of AVA_VAN | CPP_ND_V2.2E | Applicable |
| TD0555 | NIT Technical Decision for RFC Reference incorrect in TLSS Test | CPP_ND_V2.2E | Applicable |
| TD0556 | NIT Technical Decision for RFC 5077 question | CPP_ND_V2.2E | Applicable |
| TD0563 | NiT Technical Decision for Clarification of audit date information | CPP_ND_V2.2E | Applicable |
| TD0564 | NiT Technical Decision for Vulnerability Analysis Search Criteria | CPP_ND_V2.2E | Applicable |
| TD0569 | NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 | CPP_ND_V2.2E | Applicable |
| TD0570 | NiT Technical Decision for Clarification about FIA_AFL.1 | CPP_ND_V2.2E | Applicable |
| TD0571 | NiT Technical Decision for Guidance on how to handle FIA_AFL.1 | CPP_ND_V2.2E | Applicable |
| TD0572 | NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers | CPP_ND_V2.2E | Applicable |
| TD0580 | NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e | CPP_ND_V2.2E | Applicable |
| TD0581 | NIT Technical Decision for Elliptic curve-based key | CPP_ND_V2.2E | Applicable |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| | establishment and NIST SP 800-56Arev3 | | |
| TD0591 | NIT Technical Decision for Virtual TOEs and hypervisors | CPP_ND_V2.2E | Applicable |
| TD0592 | NIT Technical Decision for Local Storage of Audit Records | CPP_ND_V2.2E | Applicable |
| TD0631 | NIT Technical Decision for Clarification of public key authentication for SSH Server | CPP_ND_V2.2E | Applicable |
| TD0632 | NIT Technical Decision for Consistency with Time Data for vNDs | CPP_ND_V2.2E | Applicable |
| TD0633 | NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance | CPP_ND_V2.2E | Applicable |
| TD0635 | NIT Technical Decision for TLS Server and Key Agreement Parameters | CPP_ND_V2.2E | Applicable |
| TD0636 | NIT Technical Decision for Clarification of Public Key User Authentication for SSH | CPP_ND_V2.2E | FCS_SSHC_EXT.1 not claimed. |
| TD0638 | NIT Technical Decision for Key Pair Generation for Authentication | CPP_ND_V2.2E | Applicable |
| TD0639 | NIT Technical Decision for Clarification for NTP MAC Keys | CPP_ND_V2.2E | Applicable |
| TD0656 | Missing EAs for VPN GW Optional Headend SFRs | MOD_VPNGW_V1.2 | FTA_SSL.3/VPN not claimed. |
| TD0657 | IPSEC_EXT.1.6 GCM support for VPN GW | MOD_VPNGW_V1.2 | Applicable |
| TD0670 | NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing | CPP_ND_V2.2E | FCS_TLSC_EXT.2 not claimed. |
| TD0683 | RFC 2460 to be replaced with RFC 8200 | MOD_VPNGW_V1.2 | Applicable |
| TD0723 | Correction to ECDSA Curve Selection | MOD_VPNGW_V1.2 | Applicable |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0738 | NIT Technical Decision for Link to Allowed-With List | CPP_ND_V2.2E | Applicable |
| TD0771 | Correction to FIA_PSK_EXT.3 EA | MOD_VPNGW_V1.2 | FIA_PSK_EXT.3 not claimed. |
| TD0790 | NIT Technical Decision: Clarification Required for testing IPv6 | CPP_ND_V2.2E | FCS_TLSC_EXT.1 not claimed. |
| TD0792 | NIT Technical Decision: FIA_PMG_EXT.1 – TSS EA not in line with SFR | CPP_ND_V2.2E | Applicable |

## 1.4 Terminology

**Table 3: Terminology**

| Term | Definition |
|---|---|
| CC | Common Criteria |
| CLI | Command Line Interface |
| cPP | Collaborative Protection Profile |
| CSfC | Commercial Solutions for Classified |
| CSP | Critical Security Parameter |
| GUI | Graphical User Interface |
| IKE | Internet Key Exchange |
| IPsec | Internet Protocol Security |
| LAN | Local Area Network |
| NDcPP | collaborative Protection Profile for Network Devices |
| NTP | Network Time Protocol |
| OCSP | Online Certificate Status Protocol |
| PDF | Portable Document Format |
| PP | Protection Profile |
| RADIUS | Remote Authentication Dial-In User Service |

| Term | Definition |
|---|---|
| RAP | Aruba Remote Access Point |
| SDRAM | Synchronous Dynamic Random Access Memory |
| SNMP | Simple Network Management Protocol |
| ST | Security Target |
| TACACS+ | Terminal Access Controller Access Control System |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| VLAN | Virtual Local Area Network |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WIDS | Wireless Intrusion Detection System |

# 2 TOE Description

## 2.1 Type

4 The TOE is a distributed system of network devices comprised of Aruba Remote Access Points (RAP) and an Aruba Mobility Controller running ArubaOS 8.10. The Mobility Controller provides VPN gateway functionality for gateway to gateway VPN connections. The RAP and Mobility Controller communicate via IPsec.

5 The distributed TOE aligns with Use Case 3 per CPP_ND_V2.2E where cPP requirements cannot be fulfilled without the Management Component. To illustrate this mapping, the Management Component is fulfilled by the Aruba Mobility Controller, and the Network Device Component is fulfilled by Aruba Remote Access Points.

## 2.2 Usage

### 2.2.1 Deployment

6 The TOE is deployed in a distributed configuration with the Aruba Remote Access Points providing connectivity for wireless clients in a branch deployment, and the Aruba Mobility Controller serving as a gateway between wired and wireless networks as well as command and control functionality over Aruba RAPs.

7 Figure 1 depicts an example deployment of the TOE devices (enclosed in red).

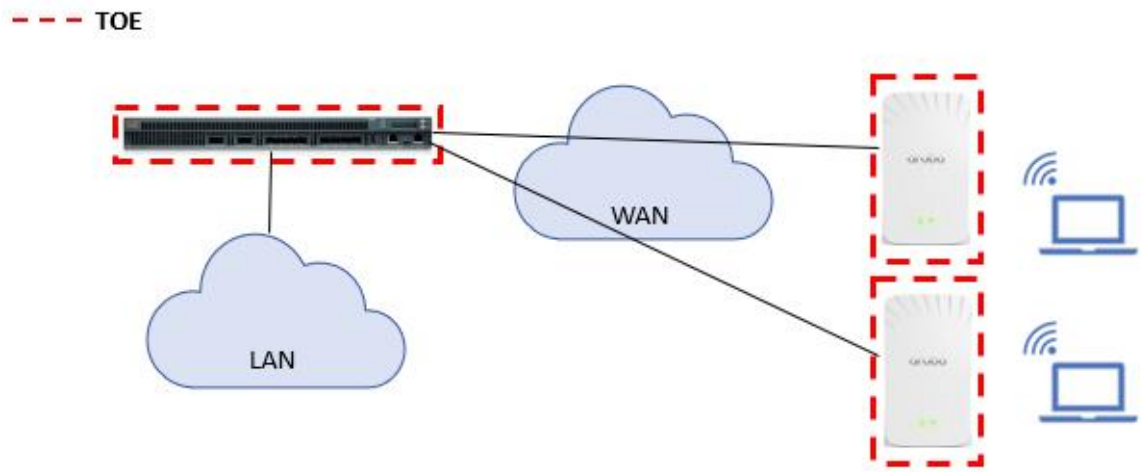


**Figure 1: Example TOE deployment**

### 2.2.2 Interfaces

8 The TOE interfaces are depicted in Figure 2.

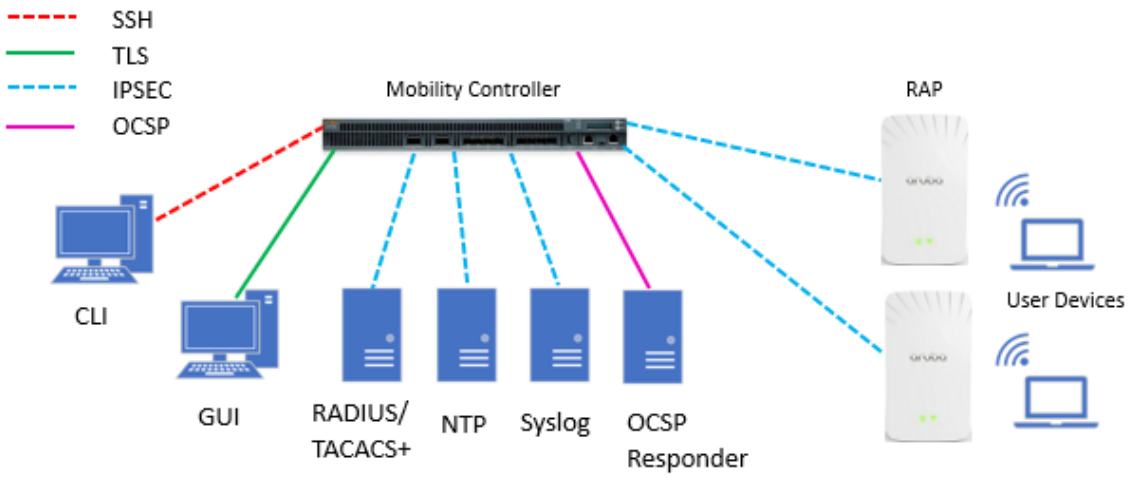


**Figure 2: TOE interfaces**

9 The TOE interfaces are as follows:

a) **CLI.** Local serial and remote SSH command line interface.
**Note:** The SSH channel can be tunneled over IPsec.

b) **GUI.** Web-based management UI via HTTPS/TLS.
**Note:** This channel can be optionally tunneled over IPsec and was tested in this evaluation.

c) **RAP IPsec.** Secure tunnel between RAP and Mobility Controller via IPsec (TOE acts as peer).

d) **RADIUS/TACACS+.** Authentication servers for user authentication via IPsec (TOE is VPN gateway).

e) **NTP.** The TOE synchronizes time with an NTP server via IPsec (TOE is VPN gateway).

f) **Syslog.** Interface for sending audit logs to remote audit server via IPSec (TOE is VPN gateway).

g) **OCSP.** The TOE receives certificate revocation status information from an external OCSP responder.

## 2.3 Security Functions / Logical Scope

10 The TOE provides the following security functions:

a) **Security Audit.** The TOE generates logs for security relevant events including startup and shutdown of the TOE and all administrative actions. Logs are stored locally on the Mobility Controller to be accessed by an administrator or can be configured to be sent via syslog to a remote server in the operational environment.

b) **Cryptographic Support.** The TOE implements key generation, establishment, and other cryptographic services to protect data in transit and at rest within the TOE. In support of cryptographic functions, the TOE implements two cryptographic modules that perform all IPsec/IKE session

operations, and functions that support all SSH, HTTPS, and TLS operations. The relevant Cryptographic Algorithm Validation Program (CAVP) certificates are listed in Table 4 with additional capabilities mappings listed in Table 18.

c) **Communication.** The TOE is a distributed configuration consisting of an Aruba Mobility Controller and Aruba Remote Access Points. The Security Administrator must enable communications between the Remote Access Points and Controller TOE components before any communication can take place. The RAPs must be configured with an appropriate RSA or ECDSA certificate and the IP address of the Aruba Mobility Controller.

d) **Identification and Authentication.** The TOE implements mechanisms to identify and authenticate administrators to ensure only authorized access to TOE functionality or TSF data is granted. These mechanisms can also be implemented through the use of RADIUS or TACACS+ servers within the operational environment.

e) **Security Management.** The TOE provides the administrator role with the capability to configure and manage all TOE security functions including cryptographic operations, user accounts, passwords, advisory banner, session inactivity, and TOE updates. The management functions are restricted to the administrator role which an administrative user must be assigned or access to these functions will be denied.

f) **Packet Filtering.** The TOE acts as a VPN gateway – a device at the edge of a private network that terminates an IPsec tunnel, which provides device authentication, confidentiality, and integrity of information traversing a public or untrusted network. The TOE provides packet filtering for gateway to gateway VPN connections. Administrators can configure security policies that determine whether to block, allow, or log a session based on traffic attributes such as source and destination port, IP address or service.

g) **Protection of the TSF.** The TOE implements a variety of protection mechanisms including authentication, self-tests, and reliable timestamping that leverages an internal hardware clock, or synchronization with an NTP server. Passwords are stored on flash using SHA1 hashes and the TOE does not provide an interface that allows for passwords or keys to be read. Confidentiality and integrity are provided for all communications between TOE components via IPsec.

h) **TOE Access.** The TOE provides session monitoring and management functions for local and remote administrative sessions. A warning banner is displayed at the management interfaces (Web GUI and CLI) to advise users on appropriate use and penalties for misuse of the system.

i) **Trusted Path/Channels.** The TOE provides secure channels between itself and local/remote administrators, including logging channels to ensure data in transit is protected. IPsec is implemented to provide encrypted channels between Mobility Controllers and third-party trusted IT entities in the operating environment. The TOE also uses IPsec to encrypt communications between TOE components and for all VPN connections. Remote Web UI access is protected with TLS/HTTPS, and CLI access is protected via SSHv2.

**Table 4: CAVP Certificates**

| Module | Services | Operational Environment | CAVP |
|---|---|---|---|
| ArubaOS Crypto Module | Provides cryptographic | Broadcom XLP432 (MIPS64) | A2689 |

| Module | Services | Operational Environment | CAVP |
|---|---|---|---|
| | functions to support all IPsec/IKE session operations | Intel Atom C3508 (Denverton) | |
| | | Qualcomm IPQ4019 (ARM Cortex-A7) | |
| | | Broadcom BCM47622L (ARM-A7) | |
| ArubaOS OpenSSL Module | Performs cryptographic functions to support all SSH, HTTPS, and TLS operations. | Broadcom XLP432 (MIPS64) | A2690 |
| | | Intel Atom C3508 (Denverton) | |
| | | Qualcomm IPQ4019 (ARM Cortex-A7) | |
| | | Broadcom BCM47622L (ARM-A7) | |
| ArubaOS Bootloader Module | Provides cryptographic functions to support firmware integrity testing. | Broadcom XLP432 (MIPS64) | A2688 |
| | | Intel Atom C3508 (Denverton) | |
| | | Qualcomm IPQ4019 (ARM Cortex-A7) | |
| | | Broadcom BCM47622L (ARM-A7) | |

## 2.4 Physical Scope

11 The physical boundary of the TOE includes the Aruba hardware components listed in Table 5 executing the ArubaOS 8.10 software. The TOE hardware is shipped to users via commercial courier and the TOE software is available via the Aruba customer portal.

**Table 5: TOE models**

| Type | Model | CPU | Software |
|---|---|---|---|
| Mobility Controller | 7210 | Broadcom XLP416 (MIPS64) | ArubaOS 8.10 |
| Mobility Controller | 7220 | Broadcom XLP432 (MIPS64) | |

| Type | Model | CPU | Software |
|---|---|---|---|
| Mobility Controller | 9004 | Intel Atom C3508 (Denverton) | |
| Remote Access Point | 303H | Qualcomm IPQ4019 (ARM Cortex-A7) | |
| Remote Access Point | 503H | Broadcom BCM47622L (ARM-A7) | |
| Remote Access Point | 505H | Broadcom BCM47622L (ARM-A7) | |

12 The ArubaOS consists of a base software package with add-on software modules that can be activated by installing the appropriate licenses. The following modules are required to be licensed and activated in the CC evaluated configuration:

a) **Policy Enforcement Firewall Next Generation.** Provides identity-based security for wired and wireless clients.

b) **Advanced Cryptography.** Required for Commercial National Security Algorithms Suite, AES-GCM, and ECDSA functionality.

13 The tested configuration included one Aruba Mobility Controller and two Aruba Remote Access Points. **Note:** This does not restrict the number of RAPs that may be managed in a conformant deployment.

### 2.4.1 Guidance Documents

14 The TOE includes the following guidance documents (PDF):

a) Aruba Common Criteria Configuration Guidance ArubaOS 8.10 Supplemental Guidance (For Aruba Remote Access Points with Mobility Controllers running ArubaOS 8.10-FIPS), Version 2.3, November 2023

b) ArubaOS 8.10.0.0 User Guide, Revision 14, 2023

c) ArubaOS 8.x Command-Line Interface Reference Guide, 2023

d) ArubaOS 8.10.0.0 Syslog Reference Guide

e) Aruba 303H Series Hospitality Access Points Installation Guide, March 2017

f) Aruba 503H Series Hospitality Access Points Installation Guide, July 2020

g) Aruba AP-505H Access Points Installation Guide, May 2023

h) Aruba 7200 Series Controller Installation Guide 0511169-06 | July 2015

i) Aruba 9004 Gateway Installation Guide, Revision 03 | June 2021

### 2.4.2 Non-TOE Components

15 The TOE operates with the following components in the environment:

a) **Audit Server.** The TOE sends audit events to a remote syslog server.

b) **NTP Server.** The TOE synchronizes time via NTP.

c) **Authentication Server.** The TOE leverages a RADIUS or TACACS+ server for handling user authentication.

d) **OCSP Responder.** The TOE receives certificate revocation status information from an external OCSP responder.

e) **Administrator Workstation.** The device(s) used by administrators to facilitate access to the TOE CLI and GUI interfaces.

### 2.4.3 Functions not included in the TOE Evaluation

16 The evaluation excludes the following Mobility Controller and RAP functionality:

a) Ability to connect with other Aruba mobility controllers;

b) SNMP (Simple Network Management Protocol) client/agent services;

c) Protocol independent multicast (routing) services for the controller;

d) 802.1X protocol;

e) Wireless IDS (WIDS);

f) Data Plane high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging);

g) Stateful firewall and deep packet inspection functions.

h) Auto-provisioning mechanisms during setup.

# 3 Security Problem Definition

17 The Security Problem Definition is reproduced from section 4 of the CPP_ND_V2.2E and section 3 of MOD_VPNGW_V1.2.

## 3.1 Threats

**Table 6: Threats - CPP_ND_V2.2E**

| Identifier | Description |
|---|---|
| T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS | Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides. |
| T.WEAK_ CRYPTOGRAPHY | Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort. |

| Identifier | Description |
|---|---|
| T.UNTRUSTED_ COMMUNICATION_ CHANNELS | Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself. |
| T.WEAK_ AUTHENTICATION_ ENDPOINTS | Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised. |
| T.UPDATE_ COMPROMISE | Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration. |
| T.UNDETECTED_ ACTIVITY | Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised. |
| T.SECURITY_ FUNCTIONALITY_ COMPROMISE | Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. |
| T.PASSWORD_ CRACKING | Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices. |
| T.SECURITY_ FUNCTIONALITY_ FAILURE | An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device. |

**Table 7: Threats - MOD_VPNGW_V1.2**

| Identifier | Description |
|---|---|
| T.DATA_INTEGRITY | Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained within the communications may be susceptible to a loss of integrity. |
| T.NETWORK_ ACCESS | Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network.<br>From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network.<br>From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link. |
| T.NETWORK_ DISCLOSURE | Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information.<br><br>From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. |

| Identifier | Description |
|---|---|
| | From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing. |
| T.NETWORK_ MISUSE | Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network.<br><br>From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services.<br><br>From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations. |
| T.REPLAY_ ATTACK | If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a “replay” attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions:<br><br>- Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome.<br><br>- No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these. |

## 3.2 Assumptions

**Table 8: Assumptions - CPP_ND_V2.2E**

| Identifier | Description |
|---|---|
| A.PHYSICAL_ PROTECTION | The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs. |
| A.LIMITED_ FUNCTIONALITY | The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).<br>If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform. |
| A.NO_THRU_ TRAFFIC_ PROTECTION | A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall). |

| Identifier | Description |
|---|---|
| A.TRUSTED_ ADMINISTRATOR | The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification). |
| A.REGULAR_ UPDATES | The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| A.ADMIN_ CREDENTIALS_ SECURE | The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside. |
| A.COMPONENTS_ RUNNING (applies to distributed TOEs only) | For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components. |
| A.RESIDUAL_ INFORMATION | The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 9: Assumptions - MOD_VPNGW_V1.2**

| Identifier | Description |
|---|---|
| A.CONNECTIONS | It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE's security policies will be enforced on all applicable network traffic flowing among the attached networks. |

## 3.3 Organizational Security Policies

**Table 10: Organizational Security Policies - CPP_ND_V2.2E**

| Identifier | 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. |

18 No additional Organizational Security Policies identified in MOD_VPNGW_v1.2.

# 4 Security Objectives

19 The following security objectives are reproduced from section 5 of the CPP_ND_V2.2E and section 4 of the MOD_VPNGW_V1.2.

## 4.1 Security Objectives for the Operational Environment

**Table 11: Security Objectives for the Operational Environment (CPP_ND_V2.2E)**

| Identifier | Description |
|---|---|
| OE.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment. |
| OE.NO_GENERAL_ PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| OE.NO_THRU_ TRAFFIC_ PROTECTION | The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment. |
| OE.TRUSTED_ADMIN | Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted. |
| OE.UPDATES | The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| OE.ADMIN_ CREDENTIALS_ SECURE | The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside. |
| OE.COMPONENTS_ RUNNING (applies to distributed TOEs only) | For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly. |
| OE.RESIDUAL_ INFORMATION | The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 12: Security Objectives for the Operational Environment (MOD_VPNGW_V1.2)**

| Identifier | Description |
|---|---|
| OE.CONNECTIONS | The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks. |

## 4.2 Security Objectives for the TOE

**Table 13: Security Objectives for the TOE (MOD_VPNGW_V1.2)**

| Identifier | Description |
|---|---|
| O.ADDRESS_ FILTERING | To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement Packet Filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) or receiving (destination) applicable network traffic as well as on established connection information.<br><br>Addressed by: FPF_RUL_EXT.1, FTA_VCM_EXT.1 (optional) |
| O.AUTHENTICATION | To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.<br><br>Addressed by: FCS_IPSEC_EXT.1 (refined from Base-PP), FIA_X509_EXT.1/Rev (from Base-PP), FIA_X509_EXT.2 (refined from Base-PP), FIA_X509_EXT.3 (from Base-PP), FTP_ITC.1/VPN, FPF_MFA_EXT.1 (optional), FTA_SSL.3/VPN (optional), FTA_TSE.1 (optional), FCS_EAP_EXT.1 (selection-based), FIA_HOTP_EXT.1 (selection-based), FIA_PSK_EXT.1 (selection-based), FIA_PSK_EXT.2 (selection-based), FIA_PSK_EXT.3 (selection-based), FIA_PSK_EXT.4 (selection-based), FIA_PSK_EXT.5 (selection-based), FIA_TOTP_EXT.1 (selection-based) |
| O.CRYPTOGRAPHIC _FUNCTIONS | To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.<br><br>Addressed by: FCS_COP.1/DataEncryption (refined from Base-PP), FCS_IPSEC_EXT.1 (refined from Base-PP), FCS_CKM.1/IKE, FCS_EAP_EXT.1 (selection-based) |

| Identifier | Description |
|---|---|
| O.FAIL_SECURE | There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.<br>Addressed by: FPT_TST_EXT.1 (refined from Base-PP), FPT_TUD_EXT.1 (refined from Base-PP), FPT_FLS.1/SelfTest, FPT_TST_EXT.3 |
| O.PORT_FILTERING | To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.<br>Addressed by: FPF_RUL_EXT.1 |
| O.SYSTEM_ MONITORING | To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).<br>Addressed by: FAU_GEN.1/VPN, FPF_RUL_EXT.1 |
| O.TOE_ ADMINISTRATION | TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.<br>Addressed by: FMT_MTD.1/CryptoKeys (refined from Base-PP), FMT_SMF.1/VPN |

20 No additional Security Objectives for the TOE identified in CPP_ND_V2.2E.

# 5 Security Requirements

## 5.1 Conventions

21 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

22 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the NDcPP.

## 5.2 Extended Components Definition

23 Refer to the Extended Components Definitions section of the PP and PP-Module as follows:

a) CPP_ND_V2.2E – Appendix 'C'

b) MOD_VPNGW_V1.2 – Appendix 'C'

## 5.3 Functional Requirements

**Table 14: Summary of SFRs**

| Requirement | Title | PP/Module Source |
|---|---|---|
| FAU_GEN.1 | Audit Data Generation | CPP_ND_V2.2E |
| FAU_GEN_EXT.1 | Security Audit Generation | CPP_ND_V2.2E |
| FAU_GEN.1/VPN | Audit Data Generation (VPN Gateway) | MOD_VPNGW_V1.2 |
| FAU_GEN.2 | User Identity Association | CPP_ND_V2.2E |
| FAU_STG_EXT.1 | Protected Audit Event Storage | CPP_ND_V2.2E |
| FAU_STG_EXT.4 | Protected Local Audit Event Storage for Distributed TOEs | CPP_ND_V2.2E |
| FAU_STG_EXT.5 | Protected Remote Audit Event Storage for Distributed TOEs | CPP_ND_V2.2E |
| FCO_CPC_EXT.1 | Component Registration Channel Definition | CPP_ND_V2.2E |
| FCS_CKM.1 | Cryptographic Key Generation | CPP_ND_V2.2E |

| Requirement | Title | PP/Module Source |
|---|---|---|
| FCS_CKM.1/IKE | Cryptographic Key Generation (for IKE Peer Authentication) | MOD_VPNGW_V1.2 |
| FCS_CKM.2 | Cryptographic Key Establishment | CPP_ND_V2.2E |
| FCS_CKM.4 | Cryptographic Key Destruction | CPP_ND_V2.2E |
| FCS_COP.1/DataEncryption | Cryptographic Operation (AES Data Encryption/Decryption) | CPP_ND_V2.2E MOD_VPNGW_V1.2 |
| FCS_COP.1/SigGen | Cryptographic Operation (Signature Generation and Verification) | CPP_ND_V2.2E |
| FCS_COP.1/Hash | Cryptographic Operation (Hash Algorithm) | CPP_ND_V2.2E |
| FCS_COP.1/KeyedHash | Cryptographic Operation (Keyed Hash Algorithm) | CPP_ND_V2.2E |
| FCS_RBG_EXT.1 | Random Bit Generation | CPP_ND_V2.2E |
| FCS_HTTPS_EXT.1 | HTTPS Protocol | CPP_ND_V2.2E |
| FCS_IPSEC_EXT.1/VPN | IPsec Protocol (VPN Gateway) | CPP_ND_V2.2E MOD_VPNGW_V1.2 |
| FCS_IPSEC_EXT.1/ITT | IPsec Protocol (Inter-TOE Communications) | CPP_ND_V2.2E |
| FCS_NTP_EXT.1 | NTP Protocol | CPP_ND_V2.2E |
| FCS_SSHS_EXT.1 | SSH Server Protocol | CPP_ND_V2.2E |
| FCS_TLSS_EXT.1 | TLS Server Protocol Without Mutual Authentication | CPP_ND_V2.2E |
| FIA_AFL.1 | Authentication Failure Management | CPP_ND_V2.2E |
| FIA_PMG_EXT.1 | Password Management | CPP_ND_V2.2E |
| FIA_UIA_EXT.1 | User Identification and Authentication | CPP_ND_V2.2E |
| FIA_UAU_EXT.2 | Password-based Authentication Mechanism | CPP_ND_V2.2E |
| FIA_UAU.7 | Protected Authentication Feedback | CPP_ND_V2.2E |

| Requirement | Title | PP/Module Source |
|---|---|---|
| FIA_X509_EXT.1/Rev | X.509 Certificate Validation | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |
| FIA_X509_EXT.1/ITT | X.509 Certificate Validation | CPP_ND_V2.2E |
| FIA_X509_EXT.2 | X.509 Certificate Authentication | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |
| FIA_X509_EXT.3 | X.509 Certificate Requests | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |
| FMT_MOF.1/ManualUpdate | Management of Security Functions Behaviour | CPP_ND_V2.2E |
| FMT_MOF.1/Services | Management of Security Functions Behavior | CPP_ND_V2.2E |
| FMT_MTD.1/CoreData | Management of TSF Data | CPP_ND_V2.2E |
| FMT_MTD.1/CryptoKeys | Management of TSF Data | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |
| FMT_SMF.1 | Specification of Management Functions | CPP_ND_V2.2E |
| FMT_SMF.1/VPN | Specification of Management Functions | MOD_VPNGW_V1.2 |
| FMT_SMR.2 | Restrictions on Security Roles | CPP_ND_V2.2E |
| FPF_RUL_EXT.1 | Packet Filtering Rules | MOD_VPNGW_V1.2 |
| FPT_FLS.1/SelfTest | Failure with Preservation of Secure State (Self-Test Failures) | MOD_VPNGW_V1.2 |
| FPT_ITT.1 | Basic Internal TSF Data Transfer Protection | CPP_ND_V2.2E |
| FPT_ITT.1/Join | Basic Internal TSF Data Transfer Protection | CPP_ND_V2.2E |
| FPT_SKP_EXT.1 | Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) | CPP_ND_V2.2E |
| FPT_APW_EXT.1 | Protection of Administrator Passwords | CPP_ND_V2.2E |
| FPT_TST_EXT.1 | TSF Testing | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |

| Requirement | Title | PP/Module Source |
|---|---|---|
| FPT_TST_EXT.3 | Self-Test with Defined Methods | MOD_VPNGW_V1.2 |
| FPT_TUD_EXT.1 | Trusted Update | CPP_ND_V2.2E<br>MOD_VPNGW_V1.2 |
| FPT_STM_EXT.1 | Reliable Time Stamps | CPP_ND_V2.2E |
| FTA_SSL_EXT.1 | TSF-initiated Session Locking | CPP_ND_V2.2E |
| FTA_SSL.3 | TSF-initiated Termination | CPP_ND_V2.2E |
| FTA_SSL.4 | User-initiated Termination | CPP_ND_V2.2E |
| FTA_TAB.1 | Default TOE Access Banners | CPP_ND_V2.2E |
| FTP_ITC.1 | Inter-TSF trusted channel | CPP_ND_V2.2E |
| FTP_ITC.1/VPN | Inter-TSF Trusted Channel (VPN Communications) | MOD_VPNGW_V1.2 |
| FTP_TRP.1/Admin | Trusted Path | CPP_ND_V2.2E |

### 5.3.1 Security Audit (FAU)

#### FAU_GEN.1 Audit Data Generation

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

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

b) All auditable events for the not specified level of audit;

*c) All administrative actions comprising:*

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- *[no other actions];*

d) *Specifically defined auditable events listed in ~~**Table 2**~~ **Table 15**.*

**Table 15: Audit Events**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None. | None. |
| FAU_GEN_EXT.1 | None. | None. |
| FAU_GEN.2 | None. | None. |
| FAU_STG_EXT.1 | None. | None. |
| FAU_STG_EXT.4 | None. | None. |
| FAU_STG_EXT.5 | None. | None. |
| FCO_CPC_EXT.1 | Enabling communications between a pair of components.<br>Disabling communications between a pair of components. | Identities of the endpoint pairs enabled or disabled. |
| FCS_CKM.1 | None. | None. |
| FCS_CKM.2 | None. | None. |
| FCS_CKM.4 | None. | None. |
| FCS_COP.1/DataEncryption | None. | None. |
| FCS_COP.1/SigGen | None. | None. |
| FCS_COP.1/Hash | None. | None. |
| FCS_COP.1/KeyedHash | None. | None. |
| FCS_RBG_EXT.1 | None. | None. |
| FCS_HTTPS_EXT.1 | Failure to establish a HTTPS session. | Reason for failure. |
| FCS_IPSEC_EXT.1/VPN | Failure to establish an IPsec SA. | Reason for failure. |
| FCS_IPSEC_EXT.1/ITT | Failure to establish an IPsec SA. | Reason for failure. |
| FCS_NTP_EXT.1 | Configuration of a new time server.<br>Removal of configured time server. | Identity of new/removed time server. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_SSHS_EXT.1 | Failure to establish an SSH session | Reason for failure |
| FCS_TLSS_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FIA_AFL.1 | Unsuccessful login attempts limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| FIA_PMG_EXT.1 | None. | None. |
| FIA_UIA_EXT.1 | All use of identification and authentication mechanism. | Provided user identity, origin of the attempt (e.g., IP address). |
| FIA_UAU_EXT.2 | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| FIA_UAU.7 | None. | None. |
| FIA_X509_EXT.1/Rev | Unsuccessful attempt to validate a certificate.<br>Any addition, replacement, or removal of trust anchors in the TOE's trust store. | Reason for failure.<br>Identification of certificates added, replaced, or removed as trust anchor in the TOE's trust store. |
| FIA_X509_EXT.1/ITT | Unsuccessful attempt to validate a certificate.<br>Any addition, replacement, or removal of trust anchors in the TOE's trust store. | Reason for failure of certificate validation.<br>Identification of certificates added, replaced, or removed as trust anchor in the TOE's trust store. |
| FIA_X509_EXT.2 | None. | None. |
| FIA_X509_EXT.3 | None. | None. |
| FMT_MOF.1/ManualUpdate | Any attempt to initiate a manual update | None. |
| FMT_MOF.1/Services | None. | None. |
| FMT_MTD.1/CoreData | None. | None. |
| FMT_MTD.1/CryptoKeys | None. | None. |
| FMT_SMF.1 | All management activities of TSF data. | None. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FMT_SMR.2 | None. | None. |
| FPT_ITT.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| FPT_ITT.1/Join | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| FPT_SKP_EXT.1 | None. | None. |
| FPT_APW_EXT.1 | None. | None. |
| FPT_TST_EXT.1 | None. | None. |
| FPT_TUD_EXT.1 | Initiation of update; result of the update attempt (success or failure) | None. |
| FPT_STM_EXT.1 | Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| FTA_SSL_EXT.1 (if “terminate the session” is selected) | The termination of a local session by the session locking mechanism. | None. |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism. | None. |
| FTA_SSL.4 | The termination of an interactive session. | None. |
| FTA_TAB.1 | None. | None. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FTP_ITC.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| FTP_TRP.1/Admin | Initiation of the trusted path.<br>Termination of the trusted path.<br>Failure of the trusted path functions. | None. |

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* ***~~Table 2~~ Table 15***.

### FAU_GEN_EXT.1 Security Audit Generation

FAU_GEN_EXT.1.1 The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

### FAU_GEN.1/VPN Audit Data Generation (VPN Gateway)

FAU_GEN.1.1/VPN The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions

b) Indication that TSF self-test was completed

c) Failure of self-test

d) All auditable events for the [*not specified*] level of audit; and

e) [*auditable events defined in the Auditable Events for Mandatory Requirements table*].

FAU_GEN.1.2/VPN The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*additional information defined in the Auditable Events for Mandatory Requirements table for each auditable event, where applicable*].

**Table 16: Auditable Events for Mandatory Requirements**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1/VPN | No events specified. | N/A |
| FCS_CKM.1/IKE | No events specified. | N/A |
| FMT_SMF.1/VPN | All administrative actions. | No additional information. |
| FPF_RUL_EXT.1 | Application of rules configured with the 'log' operation. | Source and destination addresses.<br>Source and destination ports.<br>Transport Layer protocol. |
| FPT_FLS.1/SelfTest | No events specified. | N/A |
| FPT_TST_EXT.3 | No events specified. | N/A |
| FTP_ITC.1/VPN | Initiation of the trusted channel. | No additional information. |
| FTP_ITC.1/VPN | Termination of the trusted channel. | No additional information. |
| FTP_ITC.1/VPN | Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channel establishment attempt. |

### 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.

### FAU_STG_EXT.1 Protected Audit Event 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 according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself. In addition [

- The TOE shall be a distributed TOE that stores audit data on the following TOE components: [*Mobility Controller*],

- The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: *[audit records generated by Remote Access Points are transmitted to Mobility Controller]*].

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: [*FIFO – First in, First out*], [*no other action*]] when the local storage space for audit data is full.

### FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOE's

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: *[[overwrite previous audit records according to the following rule: [first in, first out]]*].

### FAU_STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOE's

FAU_STG_EXT.5.1 Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [FPT_ITT.1].

## 5.3.2 Communication (FCO)

### FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1.1 The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- A channel that meets the secure channel requirements in [FPT_ITT.1]]

for at least *TSF data.*

FCO_CPC_EXT.1.3 The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

**Application Note:** The channel pertaining to the registration process is addressed by FPT_ITT.1/Join.

## 5.3.3 Cryptographic Support (FCS)

### FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;
- ECC schemes using ‘NIST curves’ [P-256, P-384] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;
- FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526]

]~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

### FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)

FCS_CKM.1.1/IKE The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a specified cryptographic key generation algorithm: [

- **FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 for RSA schemes;**
- **FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 for ECDSA schemes and implementing “NIST curves” P-384 and [P-256]**

**] and [**

- **No other key generation algorithm**

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

**Application Note:** This SFR has been modified by TD0723.

### FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [groups listed in RFC 3526];

] ~~that meets the following: [assignment: list of standards].~~

Application note: This SFR was changed by TD0580 and TD0581.

### FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
    - *Logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes]]*

] that meets the following: *No Standard*.

### FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [**CBC, GCM**] **and** [**CTR**] mode and cryptographic key sizes [**128 bits, 256 bits**] **and [no other cryptographic key sizes**] that meet the following: AES as specified in ISO 18033-3, [**CBC as specified in ISO 10116, GCM as specified in ISO 19772**] **and** [**CTR as specified in ISO 10116**].

Application Note: This SFR has been modified from its definition in the NDcPP to support this PP-Module's IPsec requirements by mandating support for at least one of CBC or GCM modes and at least one of 128-bit or 256-bit key sizes at minimum. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.

### FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits, 384 bits*],

] that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4]

### FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and **message digest sizes [160, 256, 384, 512] bits** that meet the following: *ISO/IEC 10118-3:2004*.

### FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384] and cryptographic key sizes *[160, 256, 384]* **and message digest sizes [160, 256, 384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

### FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*one*] software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

### FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

### FCS_IPSEC_EXT.1/VPN IPsec Protocol (VPN Gateway)

FCS_IPSEC_EXT.1.1/VPN The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2/VPN The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

FCS_IPSEC_EXT.1.3/VPN The TSF shall implement [tunnel mode].

FCS_IPSEC_EXT.1.4/VPN The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms **[AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)] and [no other algorithm]** together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-1].

Application Note: This SFR element has been modified from its definition in the NDcPP by mandating either 128 or 256 bit key sizes for AES-CBC or AES-GCM, thereby disallowing for the sole selection of 192 bit key sizes.

FCS_IPSEC_EXT.1.5/VPN The TSF shall implement the protocol: [

- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23] and [RFC 4868 for hash functions]

].

FCS_IPSEC_EXT.1.6/VPN The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms [AES-CBC-128, AES-CBC-256 (specified in RFC 3602)].

**Application Note:** This SFR element has been modified by TD0657.

FCS_IPSEC_EXT.1.7/VPN The TSF shall ensure that [

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
  - Length of time, where the time values can be configured within [*1 to 24*] hours.

  ]

].

FCS_IPSEC_EXT.1.8/VPN The TSF shall ensure that [

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [
  - Number of bytes;
  - Length of time, where the time values can be configured within [*1 to 8*] hours;

  ]

].

FCS_IPSEC_EXT.1.9/VPN 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*] bits.

FCS_IPSEC_EXT.1.10/VPN The TSF shall generate nonces used in [IKEv2] exchanges of length [

- according to the security strength associated with the negotiated DH group,
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

].

FCS_IPSEC_EXT.1.11/VPN The TSF shall ensure that IKE protocols implement DH Group(s)

- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and [**
- *[14 (2048-bit MODP)] according to RFC 3526,*

].

Application Note: This SFR element has been modified from its definition in the NDcPP by mandating DH groups 19 and 20, both of which are selectable in the original definition of the element. Any groups other than 19 and 20 may be selected by the ST author but they are not required for conformance to this PP-Module.

FCS_IPSEC_EXT.1.12/VPN 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 [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 [IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13/VPN The TSF shall ensure that all IKE protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.14/VPN The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: **Distinguished Name (DN)**, [no other reference identifier type].

Application Note: This PP-Module requires DN to be supported for certificate reference identifiers at minimum. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.

### FCS_IPSEC_EXT.1/ITT IPsec Protocol (Inter-TOE Communications)

FCS_IPSEC_EXT.1.1/ITT The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2/ITT The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

FCS_IPSEC_EXT.1.3/ITT The TSF shall implement [tunnel mode].

FCS_IPSEC_EXT.1.4/ITT The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms **[AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)] and [no other algorithm]** together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-1].

FCS_IPSEC_EXT.1.5/ITT The TSF shall implement the protocol: [

- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23] and [RFC 4868 for hash functions]

].

FCS_IPSEC_EXT.1.6/ITT The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms [AES-CBC-128, AES-CBC-256 (specified in RFC 3602)].

FCS_IPSEC_EXT.1.7/ITT The TSF shall ensure that [

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
    - Length of time, where the time values can be configured within [*1 to 24*] hours.

    ]

].

FCS_IPSEC_EXT.1.8/ITT The TSF shall ensure that [

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [
    - Length of time, where the time values can be configured within [*1 to 8*] hours;

    ]

].

FCS_IPSEC_EXT.1.9/ITT 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*] bits.

FCS_IPSEC_EXT.1.10/ITT The TSF shall generate nonces used in [IKEv2] exchanges of length [

- according to the security strength associated with the negotiated Diffie-Hellman group;
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

].

FCS_IPSEC_EXT.1.11/ITT The TSF shall ensure that IKE protocols implement DH Group(s)

[

- [14 (2048-bit MODP)] according to RFC 3526,
- [19 (256-bit Random ECP), 20 (384-bit Random ECP)] according to RFC 5114.

].

FCS_IPSEC_EXT.1.12/ITT 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 [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 [IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13/ITT The TSF shall ensure that all IKE protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.14/ITT The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [CN: IP address] and [no other reference identifier type].

### FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [NTP v4 (RFC 5905)].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

- Authentication using [SHA1] as the message digest algorithm(s);
- [IPsec] to provide trusted communication between itself and an NTP time source.

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

### FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFC(s) 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8308 section 3.1, 8332].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [password based].

Application note: This SFR was changed by TD0631.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256 kilo*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, rsa-sha2-256, rsa-sha2-512] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha1-96, hmac-sha2-256] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [ecdh-sha2-nistp256] and [ecdh-sha2-nistp384] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

### FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [TLS1.1].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [ECDHE curves [secp256r1] and no other curves].

FCS_TLSS_EXT.1.4 The TSF shall support [session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

### 5.3.4 Identification and Authentication (FIA)

#### FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-10*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

#### FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [ “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, *[“_”, “+”]];*

b) Minimum password length shall be configurable to between [*8*] and [*32] characters.*

#### FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow 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;
- [[*no other actions*]]

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.

#### FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based, SSH public key-based, [*RADIUS and TACACS+ username/password]*] authentication mechanism to perform local administrative user authentication.

#### 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**.

### FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status protocol (OCSP) as specified in RFC 6960]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
    - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

### FIA_X509_EXT.1/ITT X.509 Certificate Validation

FIA_X509_EXT.1.1/ITT The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [no revocation method]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field..

FIA_X509_EXT.1.2/ITT The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

### FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and [no other protocols],** and [no additional uses].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [not accept the certificate].

Application Note: The Base-PP allows the ST author to specify the TSF's use of X.509 certificates. Because this PP-Module mandates IPsec functionality, the SFR has been refined to force the inclusion of it. Other functions specified by the Base-PP may be chosen without restriction.

### FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

Application Note: This is specified as a selection-based SFR in the Base-PP but is mandatory for any TOE that claims conformance to this PP-Module because a conformant TOE will always have the ability to present an X.509 certificate to an external entity as part of IPsec communications. Therefore, a mechanism for the TSF to obtain a certificate for its own use is required.

## 5.3.5 Security Management (FMT)

### FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions *to perform manual updates* to *Security Administrators*.

### FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1.1/Services The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

### FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

### FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to [[*manage*]] the [*cryptographic keys and certificates used for **VPN** operation]* to *[Security Administrators]*.

Application Note: This SFR, defined in the NDcPP as selection-based, is mandated for inclusion in this PP-Module because the refinements to FMT_SMF.1 mandate its inclusion. Note that it is also refined to refer specifically to keys and certificates used for VPN operation.

### FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - Ability to start and stop services;
    - Ability to manage the cryptographic keys;
    - Ability to configure the cryptographic functionality;
    - Ability to configure the lifetime for IPsec SAs;
    - Ability to configure the interaction between TOE components;
    - Ability to set the time which is used for time-stamps;
    - Ability to configure NTP;
    - Ability to configure the reference identifier for the peer;

- Ability to manage the TOE's trust store and designate X509 v3 certificates as trust anchors;
- Ability to import X.509v3 certificates to the TOE's trust store;
- Ability to manage the trusted public keys database;]

**Application Note:** This SFR has been modified by TD0631

### FMT_SMF.1 /VPN Specification of Management Functions

FMT_SMF.1.1/VPN The TSF shall be capable of performing the following management functions: [

- *Definition of packet filtering rules;*
- *Association of packet filtering rules to network interfaces;*
- *Ordering of packet filtering rules by priority;*

[

- No other capabilities

*]*]

Application Note: This SFR defines additional management functions for the TOE beyond what is defined in the Base-PP as FMT_SMF.1. The TOE may have all management functionality implemented in the same logical interface; it is not necessary for "network device management" and "VPN gateway management" to be implemented in separate interfaces.

### FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

## 5.3.6 Packet Filtering (FPF)

### FPF_RUL_EXT.1 Rules for Packet Filtering

FPF_RUL_EXT.1.1 The TSF shall perform packet filtering on network packets processed by the TOE.

FPF_RUL_EXT.1.2 The TSF shall allow the definition of packet filtering rules using the following network protocols and protocol fields: [

- *IPv4 (RFC 791)*
  - *source address*
  - *destination address*
  - *protocol*
- *IPv6 (RFC 8200)*
  - *source address*
  - *destination address*
  - *next header (protocol)*
- *TCP (RFC 793)*
  - *source port*
  - *destination port*
- *UDP (RFC 768)*
  - *source port*
  - *destination port*

].

FPF_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with packet filtering rules: permit and drop with the capability to log the operation.

FPF_RUL_EXT.1.4 The TSF shall allow the packet filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.5 The TSF shall process the applicable packet filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: [*Administrator-defined*].

FPF_RUL_EXT.1.6 The TSF shall drop traffic if a matching rule is not identified.

**Application Note:** This SFR has been modified by TD0683

### 5.3.7 Protection of the TSF (FPT)

#### FPT_FLS.1/SelfTest Failure with Preservation of Secure State (Self-Test Failures)

FPT_FLS.1.1/SelfTest The TSF shall **shut down** when the following types of failures occur: [*failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests*].

Application Note: This SFR defines the expected TSF response to failures of the self-tests defined in the Base-PP.

#### FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [IPsec]**.

### FPT_ITT.1/Join Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1/Join The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [IPsec]**.

### FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

### FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext administrative passwords.

### FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on), periodically during normal operation] to demonstrate the correct operation of the TSF: **noise source health tests,** [

- *Software module integrity tests;*
- *Cryptographic known answer tests;*
- *Firmware integrity tests;*
- *Conditional self-tests*].

Application Note: This SFR is modified from its definition in the NDcPP by requiring noise source health tests to be performed regardless of what other testing is claimed. It is expected that the behavior of this testing will be described in the entropy documentation. Other self-tests may be defined at the ST author's discretion; note that the Application Note in the NDcPP regarding what other self-tests are expected is still applicable here.

### FPT_TST_EXT.3 TSF Self-Test with Defined Methods

FPT_TST_EXT.3.1 The TSF shall run a suite of the following self-tests [[*when loaded for execution]*] to demonstrate the correct operation of the TSF: [*integrity verification of stored executable code*].

FPT_TST_EXT.3.2 The TSF shall execute the self-testing through [*a TSF-provided cryptographic service specified in FCS_COP.1/**SigGen***].

Application Note: This requirement expands upon the self-test requirements defined in the NDcPP by specifying the method by which one of the self-tests is to be performed. "Stored TSF executable code" refers to the entire software image of the device

and not just the code related to the VPN gateway functionality defined by this PP-Module.

**FPT_TUD_EXT.1 Trusted update**

FPT_TUD_EXT.1.1 The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [the most recently installed version of the TOE firmware/software].

FPT_TUD_EXT.1.2 The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a **digital signature mechanism and [no other mechanisms]** prior to installing those updates.

Application Note: The NDcPP provides an option for how firmware/software updates can be verified but this PP-Module requires the digital signature method to be selected at minimum. Note that all other options specified in the NDcPP for this component are permitted so it is possible for the TSF to use code signing certificates to validate updates, in which case FPT_TUD_EXT.2 from the Base-PP is also included in the ST.

**FPT_STM_EXT.1 Reliable Time Stamps**

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

FPT_STM_EXT.1.2 The TSF shall [allow the Security Administrator to set the time, synchronize time with an NTP server].

### 5.3.8 TOE Access (FTA)

**FTA_SSL_EXT.1 TSF-initiated Session Locking**

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

**FTA_SSL.3 TSF-initiated Termination**

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

**FTA_SSL.4 User-initiated Termination**

FTA_SSL.4.1 The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

**FTA_TAB.1 Default TOE Access Banners**

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

### 5.3.9 Trusted path/channels (FTP)

#### FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall **be capable of using [IPsec] to provide** a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*authentication server, NTP server*]** 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 [

- *Communication with the remote authentication server;*
- *Transmitting audit records to an audit server;*
- *Synchronizing time with an NTP server;*

].

#### FTP_ITC.1/VPN Inter-TSF Trusted Channel (VPN Communications)

FTP_ITC.1.1/VPN The TSF shall **be capable of using IPsec to** provide a communication channel between itself and **authorized IT entities supporting VPN communications** 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/VPN The TSF shall permit [*the authorized IT entities*] to initiate communication via the trusted channel.

FTP_ITC.1.3/VPN The TSF shall initiate communication via the trusted channel for [remote VPN gateways or peers].

#### FTP_TRP.1 /Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall **be capable of using [IPsec, SSH, TLS, HTTPS] to** provide a communication path between itself and **authorized** 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 provides detection of modification of the channel data**.

FTP_TRP.1.2 /Admin The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

| | |
|---|---|
| FTP_TRP.1.3 /Admin | The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*. |

## 5.4 Assurance Requirements

24 The TOE security assurance requirements are summarized in Table 17.

**Table 17: Assurance Requirements**

| Assurance Class | Assurance Components |
|---|---|
| Security Target (ASE) | Conformance Claims (ASE_CCL.1) |
| | Extended Components Definition (ASE_ECD.1) |
| | ST Introduction (ASE_INT.1) |
| | Security Objectives for the operational environment (ASE_OBJ.1) |
| | Stated Security Requirements (ASE_REQ.1) |
| | Security Problem Definition (ASE_SPD.1) |
| | TOE Summary Specification (ASE_TSS.1) |
| Development (ADV) | Basic Functional Specification (ADV_FSP.1) |
| Guidance Documents (AGD) | Operational User Guidance (AGD_OPE.1) |
| | Preparative User Guidance (AGD_PRE.1) |
| Life Cycle Support (ALC) | Labelling of the TOE (ALC_CMC.1) |
| | TOE CM Coverage (ALC_CMS.1) |
| Tests (ATE) | Independent Testing – conformance (ATE_IND.1) |
| Vulnerability Assessment (AVA) | Vulnerability survey (AVA_VAN.1) |

25 In accordance with section 7.1 of the NDcPP, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

# 6 TOE Summary Specification

26 The following describes how the TOE fulfils each SFR included in section 5.

27 **Note:** Although the tested configuration included one Aruba Mobility Controller and two Aruba Remote Access Points, this does not restrict the number of RAPs that may be managed in a conformant deployment. In the evaluated configuration, each RAP implements the same security functions as shown in Table 21.

## 6.1 Security Audit

### 6.1.1 FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, & FAU_GEN_EXT.1

28 The TOE generates audit records for security relevant and other events as they occur. The events that can cause an audit record to be logged include: start-up and shutdown of the TOE; all attempts to initiate a secure communication channel; and all administrator actions comprising:

a) Administrative login and logout (including the name of the user account).

b) Enabling and disabling communications between a pair of components.

c) Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).

d) Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference is logged).

e) Resetting passwords (name of related user account is logged).

f) Attempts to initiate a TOE update.

g) Modification of the behavior of the transmission of audit data to an external IT entity.

29 The TOE's audit function cannot be stopped or started independently but rather the audit function automatically starts up when the TOE boots and automatically stops when the TOE shuts down.

30 Audit records include date and time of the event, type of event, user identity that caused the event to be generated, the outcome of the event, as well as the additional content listed in Table 15 and Table 16. For audit records involving the generating/import of, changing, or deleting of cryptographic keys, the record identifies the key via reference to the certificate or key identifier associated with the key.

31 For the establishment of IPsec sessions with authorized IT entities, the TOE generates auditable events pertaining to the initiation, termination, and failure of the trusted channel session and includes information that identifies the initiators and targets of the sessions.

32 The TOE can be configured to log packets matching a packet filtering rule by including the 'log' keyword when defining a rule.

33 The Mobility Controller will generate audit events for all security functions.

34 Remote Access Points generate audit records for the following security relevant audit events which occur on that device.

a) Shutdown of AP/auditing

b) IPsec connection failures

c) Successful/unsuccessful re-imaging

35 The Remote Access Points do not store any audit records, but rather forward all audit events securely over an IPsec connection to the Controller. The Controller locally stores the audit records and forwards the audit record in real time to an external audit server.

36 Table 15 corresponds to the audit events specified in table 2 of the NDcPP and includes the audit events specified in the NDcPP for optional and selected SFRs as selected in this ST.

37 The TOE identifies the responsible user for each event based on the specific administrator or network entity (identified by IP address) that caused the event.

### 6.1.2 FAU_STG_EXT.1

38 The TOE protects audit records in local storage from unauthorized modification or deletion. There are no CLI or GUI commands to delete or modify the local logs.

39 The TOE stores audit records locally on the Mobility Controller and can also be configured to send audit records to a trusted third-party syslog server in the operational environment. Audit records generated on the RAPs are sent to the Mobility Controller where they are stored and then forwarded to the remote syslog server. All transfers occur in real-time.

40 If a connection between the RAP and controller is lost, the logs continue to be stored in the RAP log files, which are available and buffered in flash memory. Once the connection is restored, logs can be pulled when requested from the controller via issued commands. Logs do not persist when a RAP is rebooted.

41 The TOE uses IPsec to protect the communication channel between itself and the remote syslog server and also between the Controller and the RAPs. If an external syslog server has been enabled, all audit logs are simultaneously (in real-time) written to both the local audit log on the Mobility Controller and the syslog server. The local audit logs and logs sent to a remote server are identical.

42 For audit records stored locally on MC, the maximum log size is 1.04MiB. Each log consists of three files which have an individual maximum size of 341KiB.

43 The local MC protected log storage operates using the first in, first out (FIFO) method, therefore audit logs are overwritten when the available space is exhausted.

### 6.1.3 FAU_STG_EXT.4 & FAU_STG_EXT.5

44 The Remote Access Points buffer audit records in Flash for transmission to the Mobility Controller for storage. If the buffer becomes full, previous audit records are overwritten according to a first in, first out rule.

45 The transfer of audit records to the Mobility Controller occurs in real-time over an IPsec protected channel according to FPT_ITT.1.

## 6.2 Communication

### 6.2.1 FCO_CPC_EXT.1

46 The Security Administrator must enable communications between the Remote Access Points and Controller components before any communication can take place. Administrators must use the management interfaces (Web UI or CLI) of the Mobility Controller to manually enable each RAP before it can be registered to the controller and provisioned. This registration process is facilitated by FPT_ITT.1/Join.

47 All RAPs contain a factory-installed X.509 certificate (RSA2048/SHA1). Prior to deployment, the RAPs are optionally pre-configured with an ECDSA certificate and the IP address of the Aruba Mobility Controller. The WebUI on the RAP is only temporarily enabled for this purpose.

48 The RAP will establish an initial IPsec tunnel using its factory-installed X.509 certificate (RSA2048/SHA1). The administrator enables communication with the RAP from the controller by assigning the RAP to a production group. This also forces the RAP to use the ECDSA certificate (if installed) for subsequent connections and disables the WebUI on the RAP.

49 Once the RAP is connected to the Aruba Mobility Controller via the channel described in FPT_ITT.1, the RAP downloads a configuration profile from the Aruba Mobility Controller that the Administrator configured for that RAP. The RAP can begin normal operation at this point and is immediately subject to the newly downloaded IPsec profile.

50 A Security Administrator can disable a RAP from communicating with a Controller by removing the RAP from the production AP group from a MC interface. The act of removing the RAP from an AP group is performed by removing the RAP from the allowlist (which contains the allowed AP group designation), which disables the ability for the RAP to connect to the MC.

## 6.3 Cryptographic Support

51 The TOE includes the following FIPS 140-2 Level 2 certified cryptographic modules providing supporting cryptographic functions: ArubaOS Uboot Module; ArubaOS OpenSSL Module; ArubaOS Crypto Module; and Aruba Hardware Crypto Accelerator. The ArubaOS OpenSSL Crypto Module is used for SSH/HTTPS/TLS and the ArubaOS Crypto Module is used for IPsec/IKE session cryptography. Both modules implement the low level cryptographic function in support of the protocols and run self-tests. The ArubaOS Uboot Module provides the cryptographic module integrity test on boot of the device and the Aruba Hardware Crypto Accelerator component performs Known Answer self-tests. The evaluated configuration requires that the TOE be configured in FIPS mode to ensure FIPS certified functions are used.

52 The following functions have been FIPS certified in accordance with the identified standards.

**Table 18: SFR to CAVP Mapping**

| SFR | Algorithm Capability | CAVP |
|---|---|---|
| FCS_CKM.1 Cryptographic Key Generation<br>&<br>FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication) | RSA KeyGen (FIPS Pub 186-4) (2048-bit) | A2689<br>A2690 |
| | ECDSA KeyGen (FIPS Pub 186-4) (P-256, P-384) | |

| SFR | Algorithm Capability | CAVP |
|---|---|---|
| FCS_CKM.1 Cryptographic Key Generation | FFC Schemes (NIST SP 800-56A Rev. 3, RFC 3526)<br>(DH Group 14 2048-bit) | A2689 |
| FCS_CKM.2 Cryptographic Key Establishment | Elliptic Curve-based Schemes (NIST SP 800-56A Rev 3) | A2689<br>A2690 |
| | FFC Schemes (NIST SP 800-56A Rev 3) and<br>Groups Listed in RFC 3526 | A2689 |
| FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption) | AES CBC<br>(128 and 256 bits) | A2689<br>A2690 |
| | AES GCM<br>(128 and 256 bits) | A2689<br>A2690 |
| | AES CTR<br>(128 and 256 bits) | A2690 |
| FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification) | RSA SigGen<br>(FIPS 186-4)<br>(modulus 2048 bits)<br><br>RSA SigVer<br>(FIPS 186-4)<br>(modulus 2048 bits) | A2688<br>(Firmware/Update SigVer Only)<br>A2689<br>A2690 |
| | ECDSA SigGen<br>(FIPS 186-4)<br>(256, 384 bits)<br><br>ECDSA SigVer<br>(FIPS 186-4)<br>(256, 384 bits) | A2689<br>A2690 |
| FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm) | SHA-1, SHA-256, SHA-384, SHA-512<br>(160, 256, 384, and 512 bits respectively) | A2688<br>(SHA-256 only)<br>A2689<br>A2690 |
| FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm) | HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384<br>(160, 256, and 384 bits respectively) | A2689<br>A2690 |

| SFR | Algorithm Capability | CAVP |
|---|---|---|
| FCS_RBG_EXT.1 | CTR_DRBG (AES) (256 bits) | A2690 |

### 6.3.1 FCS_CKM.1

53 For IPsec, TLS, and SSH the TOE supports cryptographic key generation for RSA schemes using key sizes of 2048 bits, ECC schemes using NIST curves P-256 and P-384. These key generation schemes are supported by both the RAP and MC for IPsec connections. The RAP and MC act as sender and receiver for these connections. RSA key generation is used for SSH host keys in the MC. The aforementioned key generation schemes are also supported by the MC as a TLS Receiver. FFC Schemes using Diffie-Hellman group 14 key sizes of 2048-bit is supported for IPsec as sender and receiver by both the RAP and MC.

### 6.3.2 FCS_CKM.1/IKE

54 For keys used for IKE peer authentication, the TOE supports cryptographic key generation for RSA schemes using key sizes of 2048 bits as specified in FIPS PUB 186-4 Appendix B.3, and ECC schemes using NIST curves P-256 and P-384 as specified in FIPS PUB 186-4 Appendix B.4.

### 6.3.3 FCS_CKM.2

55 The TOE performs key establishment by generating DH parameters over NIST curves secp256r1 and secp384r1 for SSH, TLS, and IPsec. For IPsec, the TOE also supports Key establishment scheme using Diffie-Hellman MODP group 14 (2048-bit) that meets RFC 3526, Section 3.

### 6.3.4 FCS_CKM.4

56 The TOE uses the following secret keys, private keys and CSPs.

**Table 19: Key Usage**

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| DRBG entropy input | SP800-90a CTR_DRBG (512 bits) | Entropy inputs to the DRBG function used to construct the DRBG seed. | Stored in SDRAM memory (plaintext) | Zeroized by rebooting the module |
| DRBG seed | SP800-90a CTR_DRBG (384-bits) | Input to the DRBG that determines the internal state of the DRBG. Generated using DRBG derivation function. | Stored in SDRAM memory (plaintext) | Zeroized by rebooting the module |

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| DRBG Key | SP800-90a CTR_DRBG (256 bits) | This is the DRBG key used for SP800-90a CTR_DRBG. | Stored in SDRAM memory (plaintext) | Zeroized by rebooting the module |
| Diffie-Hellman private key | Diffie-Hellman Group 14 (224 bits) | Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| Diffie-Hellman public key | Diffie-Hellman Group 14 (2048 bits) | Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| Diffie-Hellman shared secret | Diffie-Hellman Group 14 (2048 bits) | Established during Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| EC Diffie-Hellman private key | EC Diffie-Hellman (Curves: P-256 or P-384). | Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| EC Diffie-Hellman public key | EC Diffie-Hellman (Curves: P-256 or P-384). | Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| EC Diffie-Hellman shared secret | EC Diffie-Hellman (Curves: P-256 or P-384) | Established during EC Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| RADIUS server shared secret | 8-128 characters shared secret | Entered by administrator. Used for RADIUS server authentication. | Stored in Flash memory (plaintext) | Zeroized by using command 'write erase all' or by overwriting with a new secret |
| Enable secret | 8-32 characters password | Entered by administrator. Used for authentication. | Stored in Flash memory (plaintext) | Zeroized by using command 'write erase all' or by overwriting with a new secret |
| User Password | 8-32 characters password | Entered by administrator. Used for User authentication. | Stored in Flash memory (plaintext) | Zeroized by using command 'write erase all' or by overwriting with a new secret |
| SKEYSEED | Shared Secret (160/256/384 bits) | A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv2) and it will be used for deriving IKE session authentication key. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| IKE session authentication key | HMAC-SHA-1/256/384 (160/256/384 bits) | The IKE session (IKE Phase I) authentication key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv2). Used for IKEv2 payload integrity verification. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| IKE session encryption key | AES (128/256 bits, CBC) | The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv2). Used for IKE payload protection. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| IPSec session encryption key | AES (128/256 bits) CBC & GCM | The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv2). Used to secure IPsec traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| IPSec session authentication key | HMAC-SHA-1 (160 bits) | The IPsec (IKE Phase II) authentication key. This key is derived via using the KDF defined in SP800-135 KDF (IKEv2). Used to verify the integrity of IPsec traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| SSHv2 session key | AES (128/256 bits) CBC & CTR | This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to secure SSHv2 traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| SSHv2 session authentication key | HMAC-SHA-1, (160-bit) | This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to verify the integrity of SSHv2 traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| TLS pre-master secret | 48 bytes secret | This key is transferred into the module, protected by TLS RSA public key. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| TLS session encryption key | AES (128, 256 bits) CBC & GCM | This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to secure TLS traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| TLS session authentication key | HMAC-SHA-1/256/384 (160/256/384 bits) | This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to verify the integrity of TLS traffic. | Stored in SDRAM memory (plaintext). | Zeroized by rebooting the module |
| RSA Private Key | RSA 2048 bit private key | This key is generated in the module. Used for IKEv2, TLS, and OCSP (signing OCSP messages). | Stored in Flash memory (Encrypted using 3DES). | Zeroized by using command 'write erase all' |
| RSA public key | RSA 2048 bits public key | This key is generated in the module. This Key can also be entered by the administrator via SSH (CLI) and/or TLS (for the GUI). Used for IKEv2, TLS, and OCSP (verifying OCSP messages). | Stored in Flash memory (plaintext). | The zeroization requirements do not apply to this key as it is a public key. |
| ECDSA Private Key | ECDSA suite B P-256 and P-384 curves | This key is generated in the module. Used for IKEv2, and TLS. | Stored in Flash memory (Encrypted using 3DES). | Zeroized by using command 'write erase all'. |
| ECDSA Public Key | ECDSA suite B P-256 and P-384 curves | This key is generated in the module. This Key can also be | Stored in Flash memory (plaintext). | The zeroization requirements do not apply to |

| CSP | CSP Type | Generation and Use | Storage | Zeroization |
|---|---|---|---|---|
| | | entered by the administrator via SSH (CLI) and/or TLS (for the GUI).<br>Used for IKEv2, and TLS. | | this key as it is a public key. |
| 3DES Key Encrypting Key (KEK) | 168-bit 3DES key | Used to protect/obfuscate RSA & ECDSA private keys. | Stored in Flash memory (plaintext) | Not subject to zeroization requirements as it is not considered a CSP. |
| Aruba Factory Public Key | RSA<br>(2048 bits) | This is RSA public key. Loaded into the module during manufacturing. Used for Firmware verification. | Stored in firmware image (plaintext). | The zeroization requirements do not apply to this key as it is a public key. |

57 The TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. This function has also been subject to FIPS 140 certification. The table above identifies all secret and private keys and Critical Security Parameters (CSPs), the related zeroization procedures and whether any interface is available to view the plaintext key.

58 Zeroization of CSPs stored in plaintext is accomplished by overwriting with all zeroes. The CLI command *'write erase all'* can be used to zeroize the keys and CSPs stored in flash memory as identified in Table 19 above. This command erases the running system configuration file and all databases including the keys and CSPs.

**Note:** The *'write memory'* command must be executed for the changes to be retained across system reboots.

### 6.3.5 FCS_COP.1/DataEncryption

59 The TOE provides encryption and decryption capabilities using AES in 128-bit and 256-bit CBC, GCM, and CTR modes. AES is implemented in the TLS, SSH, and IPSec protocols.

60 The relevant NIST CAVP certificate numbers are listed in Table 4.

### 6.3.6 FCS_COP.1/SigGen

61 The TOE supports RSA (modulus 2048) and ECDSA with elliptical curve size 256 or 384 bits for signature generation and verification.

### 6.3.7 FCS_COP.1/Hash

62 The TOE performs SHA-1, SHA-256, SHA-384, and SHA-512 cryptographic hashing services in accordance with ISO/IEC 10118-3:2004. The SHA hash algorithm is used as part of HMAC, but is also used as part of RSA digital signature creation and verification.

### 6.3.8 FCS_COP.1/KeyedHash

63 The TOE performs keyed-hash message authentication that meets the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2. The key length, hash function used, block size, and output MAC lengths are identified in the table below.

**Table 20: HMAC Characteristics**

| Algorithm | Block Size | Key Size | Digest Size |
|---|---|---|---|
| HMAC-SHA-1 | 512 bits | 160 bits | 160 bits |
| HMAC-SHA-256 | 512 bits | 256 bits | 256 bits |
| HMAC-SHA-384 | 1024 bits | 384 bits | 384 bits |

64 The relevant NIST CAVP certificate numbers are listed in Table 4.

### 6.3.9 FCS_RBG_EXT.1

65 The TOE uses a software based random bit generator that complies with AES-256 CTR_DRBG when operating in FIPS mode. AES-256 is used in conjunction with a minimum of 256 bits of entropy accumulated from one software based noise source that includes the following:

a) Timing variances over computation operations.

b) Timing variances over memory accesses.

66 The entropy value provided by these sources, combined with a NIST vetted SHA3-256 conditioning operation, suffices the minimum requirements for a FIPS approved Jitter entropy implementation that is used to seed the CTR_DRBG which is leveraged by the MC and RAP components.

### 6.3.10 FCS_HTTPS_EXT.1

67 The TOE supports TLS/HTTPS in the web-based GUI to provide secure administrator sessions. The HTTPS protocol complies with RFC 2818 and is implemented using TLS.

68 The TOE Web UI operates on an explicit port designed to natively communicate over TLS and does not attempt STARTTLS or other multi-protocol negotiations as described in section 2.3 of RFC 2818. The TOE will attempt to send closure alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818.

### 6.3.11 IPsec

#### 6.3.11.1 FCS_IPSEC_EXT.1/VPN

69 The TOE implements IPsec in accordance with RFC 4301. The TOE supports IPsec for tunnel mode.

70 The IPsec ESP protocol is implemented in conjunction with AES-CBC-128 and AES-CBC-256 (as specified by RFC 3602) together with the following truncated versions of SHA-based HMAC algorithms: HMAC-SHA-1 and with AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106).

71 The TOE implements IKEv2, with support for NAT traversal, as defined in RFC 5996 and RFC 4868 for hash functions which include HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-384. Diffie-Hellman (DH) Groups 14, 19, and 20 are supported for

IKEv2 as are RSA and ECDSA certificates. The TOE uses the AES-CBC-128 and AES-CBC-256 algorithms as specified in RFC 3602 to encrypt the IKEv2 payload.

72 The TOE generates the secret value x used in the IKEv2 Diffie-Hellman key exchange (“x” in g^x mod p) using the FIPS validated RBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256, and 384 bits (for DH Groups 14, 19, and 20, respectively).

73 The DRBG described in FCS_RBG_EXT.1 is used to randomly generate each nonce used in IKEv2 exchanges according to the security strength associated with the negotiated DH group—128 bits for Groups 14 and 19, and 192 bits for Group 20. The nonces generated are at least half the output size of the negotiated pseudorandom function (PRF) hash.

74 Lifetimes for IKEv2 SAs are established during configuration of the IKE policies by specifying the number of seconds for the SA lifetime. IKEv2 SA lifetime and volume limits can be configured via the CLI function by an authorized administrator. IKEv2 can be configured with 1-24 hours for the IKEv2 IKE_SA and within 1-8hrs for the IKEv2 IKE_CHILD SA. In the IKEv2 IKE_SA and IKE_CHILD exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE initiates IKE negotiation, the DH group is sent in order according to the peer’s configuration. When the TOE receives an IKE proposal, it will select the first match and the negotiation will fail if there is no match.

75 The TOE supports the ability to ensure by default that the strength of the negotiated symmetric algorithm in the IKEv2 CHILD_SA is less than or equal to the strength of the IKEv2 IKE_SA. As such, it is the responsibility of the Administrator to properly configure the algorithms in the IPsec policy. The symmetric algorithms supported for IKEv2 IKE_SA use the same or greater key length as the symmetric algorithms used to protect IKEv2 CHILD_SA (128-bit, and 256-bit). The available options ensure that the IKEv2 IKE_SA symmetric algorithm key length is equal to or greater than the IKEv2 CHILD_SA symmetric algorithm key length.

76 The TOE will only establish a trusted IPsec channel if the presented identifier in the received certificate matches the configure reference identifier. For IPsec connections where the TOE is acting as a VPN Gateway, the presented and reference identifiers are of the following type: Distinguished Name (DN). Fields within the DN are not individually selectable; the DN must be an exact match for the entire DN string.

77 The TOE implements an SPD and processes packets to satisfy the behavior of DISCARD, BYPASS, and PROTECT packet processing as described in RFC 4301 to determine what traffic gets protected with IPsec, what gets bypassed, and what gets dropped. Each packet is either PROTECTed using IPsec security services, DISCARDed, or allowed to BYPASS IPsec protection, based on the applicable SPD policies. The SPD is achieved via tunneling, packet filtering policies and packet filtering rules. The TOE administrator implicitly configures the IPsec SPD rules using policies. The TOE compares packets against the configured rules to determine if any of the packets match the rules. The packets can be matched based upon source IP address, destination IP address, protocol type (e.g. TCP, UDP, ICMP). Traffic not matching any rule is passed to the next stage of processing. The TOE includes a final entry in the SPD that matches anything that is otherwise unmatched, and discards it. Packet processing is described in section 6.6.

78 Aruba uses an automated test tool (Ixia IxANVL) to test conformance with RFCs.

#### 6.3.11.2 FCS_IPSEC_EXT.1/ITT

79 The inter-TOE (MC to RAP) IPsec implementation is the same as described above with the following exceptions:

a) The reference identifiers are of the following type: CN: IP address

b) The route-based SPD for inter-TOE connections applies to all traffic and enforces either “default PROTECT” or “BYPASS only to facilitate IKE traffic” operations

c) The following algorithm suites are supported based upon the certificate in use:

**RSA**

i) Phase 1: DH Group 14, AES CBC 128/256, HMAC SHA2 256

ii) Phase 2 (ESP): AES CBC 128/256, HMAC SHA1

**ECDSA P-256**

iii) Phase 1: DH Group 19, AES CBC 128/256, HMAC SHA2 256

iv) Phase 2 (ESP): AES GCM 128/256

**ECDSA P-384**

v) Phase 1: DH Group 20, AES CBC 128/256, HMAC SHA2 384

vi) Phase 2 (ESP): AES GCM 256

d) Child SA lifetimes are based on length of time only.

### 6.3.12 FCS_NTP_EXT.1

80 The TOE supports NTP v4 and can synchronize with at least three NTP time sources. The TOE updates its system time using IPsec to provide trusted communication between itself and an NTP time source; and SHA1 authentication as the message digest algorithm. The TOE does not update NTP timestamps from Broadcast and multicast addresses. The use of IPsec and SHA-1 message digest algorithm ensures the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained.

### 6.3.13 FCS_SSHS_EXT.1

81 The TOE implements the SSH protocol in accordance with RFC’s 4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668, 8308 section 3.1, and 8332.

82 Remote administration via the Command Line Interface (CLI) is protected using SSHv2. The TOE supports SSHv2 with AES (CBC and CTR) 128 or 256 bit ciphers, in conjunction with HMAC-SHA-1, HMAC-SHA1-96, and HMAC-SHA2-256 using the following key exchange methods: ecdh-sha2-nistp256 and ecdh-sha2-nistp384. The TOE also supports ssh-rsa, rsa-sha2-256, and rsa-sha2-512 for server authentication. While other ciphers and hashes are implemented in the product, they are disabled while the TOE is operating in FIPS mode.

83 The TOE supports both public-key (SSH-RSA) and password-based client authentication and can be configured. The TOE SSH connections are rekeyed after a period of no longer than 60 minutes or 512 MB of transmitted data. Both of these thresholds are checked by the TOE and rekeying is performed upon reaching the threshold that is hit first. The TOE limits packets to 256k bytes. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256k bytes) the packet will be dropped.

84 During authentication, the TOE establishes a user identity by either verifying that the SSH client’s current public key matches the one stored within the TOE’s SSH

authorized keys file, or by confirming the validity of the presented username and matching password within its database or via a configured RADIUS or TACACS+ server.

### 6.3.14 FCS_TLSS_EXT.1

85 The TOE restricts the use of TLS protocols to version 1.2 exclusively. All other SSL/TLS versions are therefore not supported by the TOE and such connection attempts are rejected.

86 Remote administration via the Web GUI is protected using TLS/HTTPS. The following cipher suites are implemented by the TOE, no configuration is needed to support any of these ciphers:

a) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

b) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

c) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

d) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

e) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

f) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

g) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

h) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

i) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

j) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

87 The key agreement parameters of the server key exchange message are specified in the RFC 5246 (section 7.4.3) for TLSv1.2.

88 The TOE performs key establishment for TLS using ECDHE curves secp256r1.

89 The TOE supports session resumption based on session IDs according to RFC 5246 and session tickets according to RFC 5077.

90 Session resumption is based on a single context and operates according to the applicable RFCs. Sessions can be reused providing all session properties are still valid and parameters are otherwise not accepted by the TOE. If the latter occurs, a full handshake process will be performed before a connection is established.

91 Session tickets are protected by implementing symmetric encryption algorithms as described in FCS_COP.1/DataEncryption and section 6.3.5. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm. Session tickets adhere to the structural format provided in section 4 of RFC 5077.

## 6.4 Identification and Authentication

### 6.4.1 FIA_AFL.1

92 After an administrator specified number of consecutive failed authentication attempts between 1 and 10 that occur in a three minute period, the TOE will lockout the offending remote administrator and log the event. The duration in time that a user is locked out upon crossing the lock out threshold is 0-60 minutes. The offending

administrator will remain locked out until the administrator configured lock-out period has expired. FIA_AFL.1 is enforced for password-based authentication by the TOE. An administrator with a public-key will never be locked out since public-key-based authentication is not subject to the password lock-out function.

### 6.4.2 FIA_PMG_EXT.1

93 The TSF enforces password management capabilities that require administrative passwords to be composed with the following characteristics:

a) Upper and lower case letters;

b) Numbers;

c) And the following special characters: ! @ # $ % ^ & * ( ) _ +

94 The minimum password length is configurable by an administrator to between 8 and 32 characters.

### 6.4.3 FIA_UIA_EXT.1

95 Prior to requiring the non-TOE entity to initiate the identification and authentication process, the TOE displays an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE (FTA_TAB.1). The TOE requires an administrator to be successfully identified and authenticated before being presented with the administration console and allowing any additional TSF-mediated actions to be executed on behalf of that user.

96 Once the TOE is operational, administrators can access the TOE interfaces through the Mobility Controller via the web GUI (HTTPS/TLS) and CLI (SSH). The logon process is initiated by the administrator via the desired interface where an authentication challenge is presented to the administrator. If the credentials entered by the administrator are valid, the authentication sequence will complete successfully and the administrator is presented with the administration interface. Additional information on the supported authentication mechanisms per interface, see section 6.4.4 (FIA_UAU_EXT.2).

97 Administrators cannot log in to the RAP since those interfaces are disabled after initial configuration.

### 6.4.4 FIA_UAU_EXT.2

98 The TOE provides local accounts utilizing password-based or public-key-based authentication and can also be configured to utilize a RADIUS or TACACS+ authentication server in its operational environment. The administrator can configure the TOE to provide the same or different authentication mechanism (local, remote) for non-administrator users and administrators. The TOE invokes the correct authentication mechanism as configured by the administrator. Local administration is performed via serial console. Authentication via the local CLI (physical connection) can be performed via username and password. Remote (SSH) authentication can be performed via username-password or public-key-based (or a combination of the two). Remote Web GUI (HTTPS/TLS) authentication can be username-password, or RADIUS, TACACS+. This description applies to the Mobility Controller. Administrator cannot log in to the RAP since the interfaces are disabled after initial configuration.

### 6.4.5 FIA_UAU.7

99 The TOE provides only obscured feedback to the administrative user while authentication is in progress at the local console by displaying no echoed characters when characters are entered.

### 6.4.6 FIA_X509_EXT.1/Rev

100 The TOE performs X.509 certificate validation at the following points:

a) On load of certificate responses

b) When processing OCSP responses

c) During IPsec peer authentication

101 In all scenarios, certificates are checked for several validation characteristics:

a) If the certificate ‘notAfter’ date is in the past, then this is an expired certificate which is considered invalid;

b) The certificate chain must terminate with a trusted CA certificate designated as a trust anchor;

c) A trusted CA certificate is defined as any certificate loaded into the TOE trust store that has, at a minimum, a basicConstraints extension with the CA flag set to TRUE;

d) The TOE validates the extendedKeyUsage field as follows:

i) TLS server certificates must have the Server authentication purpose in the extendedKeyUsage field

ii) OCSP certificates must have the OCSP signing purpose in the extendedKeyUsagefield

102 Certificate revocation checking is performed when certificates are presented to the TOE and when loaded into the TOE. Revocation status is checked using OCSP as specified in RFC 6960. All certificates in the chain except for the root are verified in order, starting with the peer cert and ending at the penultimate CA certificate.

103 The X.509 certificates for each of the given scenarios are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows:

a) The public key algorithm and parameters are checked

b) The current date/time is checked against the validity period revocation status is checked

c) Issuer name of X matches the subject name of X+1

d) Name constraints are checked

e) Policy OIDs are checked

f) Policy constraints are checked; issuers are ensured to have CA signing bits

g) Path length is checked

h) Critical extensions are processed

104 If at any point a certificate under review fails the trust chain verification activity or revocation status check, then the entire trust chain is deemed untrusted.

### 6.4.7 FIA_X509_EXT.1/ITT

105 Certificates used in authentication of distributed TOE communication (between AP and MC) are validated as described in Section 6.4.6 above using a minimum certificate chain path of two. These channels leverage the IPsec protocol to establish secure connections. Per the NDcPP, revocation checking is optional due to the additional requirements surrounding the enabling and disabling of the ITT channel as

defined in FCO_CPC_EXT.1. No revocation checking is performed for ITT connections.

### 6.4.8 FIA_X509_EXT.2

106 The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec.

107 Initially, each RAP device contains only one end point certificate. In these cases, the installed certificate is used. If a custom certificate is installed on the RAP, the TOE will use that certificate until it is removed. If a custom certificate is removed or has not been installed, the default certificate (pre-installed on all RAPs) is used.

108 For IPSec connections, the TOE uses certificates to establish a VPN tunnel between Aruba Mobility Controllers and Remote Access Points. The Mobility Controller maintains a trust store where root CA and intermediate CA certificates can be stored. This trust store is not cached and if a certificate is deleted, it is immediately untrusted. When certificates are added to the trust store, they are immediately trusted for it's given scope. The Mobility Controller maintains certificates in the trust store that are used to identify itself to remote IPsec peers. Administrators can assign certificates to specific IPsec channels.

109 When a connection cannot be established to determine the validity of a certificate, the certificate is not accepted.

110 No other distinctions are made between trusted channels, and certificate validation is performed in the same manner across trusted channels.

### 6.4.9 FIA_X509_EXT.3

111 The TOE generates Certificate Request Messages and includes the following information: public key, common name, organization, organizational unit, country. Upon receiving the CA Certificate response, the TOE will validate the chain of certificates from the Root CA.

## 6.5 Security Management

### 6.5.1 FMT_MOF.1/ManualUpdate

112 Only Security Administrators can initiate manual updates. Security Administrators must load candidate updates first onto the MC. Once the update is successfully loaded, the next time a RAP component connects to the MC, the update is downloaded to the RAP (assuming the digital signature is valid).

### 6.5.2 FMT_MOF.1/Services

113 Only Security Administrators can enable and disable, start and stop TOE functions and services.

### 6.5.3 FMT_MTD.1/CoreData

114 Only authorized Security Administrators can manage TSF data. There are no security functions available through any interfaces prior to successful administrator login. Access to TOE functionality is restricted until a Security Administrator has been identified and authenticated, at which point they will be presented with the management console and can then perform all administrative tasks including those listed in section 6.5.5.

### 6.5.4 FMT_MTD.1/CryptoKeys

115 Only Security Administrators have the capability to modify, delete, generate, and import the cryptographic keys and certificates used for VPN operation.

### 6.5.5 FMT_SMF.1 & FMT_SMF.1/VPN

116 The TOE provides the administrator with local and remote interfaces to manage all security functions identified in this Security Target, including the following:

a) Configure the access banner warnings

b) Configure the session inactivity time before session termination

c) Perform TOE updates, query TOE version, and verify the updates using digital signature capability

d) Configure the interaction between TOE components

e) Configure the authentication failure settings

f) Configure the cryptographic functionality including modifying, deleting, generating and importing cryptographic keys and certificates for VPN operation

g) Configure IPsec functionality including configuring the lifetime for IPsec SAs

h) Manage the TOE's trust store and designate X509 v3 certificates as trust anchors

i) Configure and Import X.509v3 certificates to the TOE's trust store

j) Set the time by configuring NTP services used for timestamps

k) Set the time manually

l) Manage the trusted public keys database

m) Configure the reference identifier for the peer

n) Start and stop services

117 All of the above functions can be performed on the Mobility Controller using either the CLI or the WebUI.

118 A default administrator account is configured during initial configuration where the password is set by the admin.

119 The interaction of TOE components can be configured via the Mobility Controller per FCO_CPC_EXT.1 and as described in section 6.2.1.

### 6.5.6 FMT_SMR.2

120 The TOE provides the administrator role that corresponds to the Security Administrator role specified in the NDcPP. The administrator can manage all aspects of the TOE locally or remotely using the CLI or remotely through the GUI.

## 6.6 Packet Filtering

### 6.6.1 FPF_RUL_EXT.1

121 The Aruba Mobility Controllers act as VPN gateways – devices at the edge of a private network that terminate IPsec tunnels, which provide device authentication, confidentiality, and integrity of information traversing a public or untrusted network.

122 The TOE implements the IPsec protocol and supports the following protocols:

- RFC 791 (Ipv4)
- RFC 8200 (Ipv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

123 The Aruba Quality Assurance (QA) team performs protocol compliance testing using standards based tools and interoperability testing using a range of external vendor equipment.

124 The TOE blocks the following protocols by default:

- IPv6
  - Protocol 135
  - Protocol 140

125 All other protocols are supported and allowed during normal operation of the TOE.

126 The TOE allows the definition of packet filtering policy rules based on the following attributes that are used to define rules (based on the actions: permit, deny or log) for the associated protocols:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- IPv6
  - Source address
  - Destination Address
  - Next Header (Protocol)
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

127 An authorized administrator can define the traffic that needs to be protected by configuring access-lists. The permit, deny and log operations can be associated with rules in the access-lists. Only a single access-list may be applied to a distinct Ethernet interface. Each rule can identify the following actions: permit, deny, and log. Traffic may be selected on the basis of the source and destination address, and optionally the Layer 4 protocol and port. Rules are enforced based on the order defined by the administrator in a first match basis. Packets that do not match a rule are then by default handled as configured by the administrator to drop/deny. The access lists can be applied to all network interfaces.

128 The algorithm applied to incoming packets is as follows:

- Check for IP fragments and assemble.

- Parse and identify protocol in the IP packet.
- Consult the state table to determine whether packets are part of an established session.
- Perform length checks and apply default rules (the default rules are not covered in the scope of evaluation).
- Enforce interface access-lists (ACLs) if configured.
- Derive role for the user and apply role based ACLs. If no role ACLs, then apply default ACLs (deny).
- Perform bandwidth contract enforcement.
- Perform NATing if required.

129 All packet level processing and enforcement is performed within the data plane, which is the first component that is initialized. Network interfaces are not brought 'up' until the data plane initialization is complete. Therefore, packets cannot flow during this process. In case of system error, packets are dropped by default.

130 The TOE implements a full stateful firewall instance for each user to provide granular control over user access, and separation between user classes. Stateful sessions are maintained by the TOE via a state table to determine whether packets are part of an established session.

## 6.7 Protection of the TSF

### 6.7.1 FPT_FLS.1/SelfTest

131 In order to prevent entering an insecure state, the TOE will shut down when the following failures occur: failure of power on self-tests, failure of integrity check of the TSF executable image, and failure of noise source health tests. HW based noise sources are used with Random bit generation and CTR_DRBG (AES). The Continuous Random Number Generator Test, Firmware integrity check and other power on self-tests are described below in Section 6.7.5.

### 6.7.2 FPT_ITT.1 & FPT_ITT.1/Join

132 All data transmitted between TOE components is protected from disclosure and modification by using IPsec. IPsec VPN tunnels are established between Aruba Remote Access Points and Aruba Mobility Controllers.

133 The registration channel facilitated by FPT_ITT.1/Join is also protected by using IPsec. Aruba Remote Access Points initiate communication with Aruba Mobility Controllers using IPsec VPN tunnels.

### 6.7.3 FPT_SKP_EXT.1

134 The TOE provides no interfaces that allow pre-shared, symmetric or private keys to be read. Section 6.3 describes how the pre-shared keys, symmetric keys and private keys are stored.

### 6.7.4 FPT_APW_EXT.1

135 Passwords are not stored in plaintext but rather stored in flash using a SHA1 hash. The TOE does not provide any interfaces to view passwords.

### 6.7.5 FPT_TST_EXT.1

136 The Mobility Controller and Remote Access Points run a suite of self-tests during power-up, which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. The Mobility Controller and Remote Access Points run the suite of FIPS 140-2 validated cryptographic module self-tests during start-up or reboot. Conditional self-tests are also run during the course of normal operation.

137 The following tests are performed on both MC and RAP:

- ArubaOS OpenSSL Module:
  - AES Known Answer Tests (KAT)
  - Triple-DES KAT
  - RNG KAT
  - RSA KAT
  - ECDSA (sign/verify)
  - SHA (SHA1, SHA256 and SHA384) KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256 and HMAC-SHA384) KAT
- ArubaOS Cryptographic Module
  - AES KAT
  - Triple-DES KAT
  - SHA (SHA1, SHA256, SHA384) KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384) KAT
  - RSA (sign/verify)
  - ECDSA (sign/verify)
- ArubaOS Uboot BootLoader Module
  - Firmware Integrity Test: RSA 2048-bit Signature Validation
- Aruba Hardware Crypto Accelerator Known Answer Tests:
  - AES KAT
  - AES-GCM KAT
  - Triple DES KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384) KAT

138 The following Conditional Self-tests are performed by the TOE:

- **Continuous Random Number Generator Test** –This test is run upon generation of random data by the switch's random number generators to detect failure to a constant value. The module stores the first random number for subsequent comparison, and the module compares the value of the new random number with the random number generated in the previous round and enters an error state if the comparison is successful.

- **Noise Source Health** – This test continuously measures the local frequency of occurrence of a sample value in a sequence of noise source samples to determine if the sample occurs too frequently. Thus, the test is able to detect when some value begins to occur much more frequently than expected, given the source's assessed entropy per sample.
- **Bypass test**. Ensures that the system has not been placed into a mode of operation where cryptographic operations have been bypassed, without the explicit configuration of the cryptographic officer. To conduct the test, a SHA1 hash of the configuration file is calculated and compared to the last known good hash of the configuration file. If the hashes match, the test is passed. Otherwise, the test fails (indicating possible tampering with the configuration file) and the system is halted.
- **RSA Pairwise Consistency test**. When the TOE generates a public and private key pair, it carries out pair-wise consistency tests for both encryption and digital signing. The test involves encrypting a randomly-generated message with the public key. If the output is equal to the input message, the test fails. The encrypted message is then decrypted using the private key and if the output is not equal to the original message, the test fails. The same random message is then signed using the private key and then verified with the public key. If the verification fails, the test fails.
- **ECDSA Pairwise Consistency test.** See above RSA pairwise consistency test description.
- **Firmware Load Test.** This test is identical to the Uboot BootLoader Module Firmware Integrity Test, except that it is performed at the time a new software image is loaded onto the system. Instead of being performed by the BootLoader, the test is performed by the ArubaOS operating system. If the test fails, the newly loaded software image will not be copied into the image partition, and instead will be deleted.

139 Known-answer tests (KAT) involve operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test shall fail.

140 If a self-test fails, the TOE will immediately halt operation and enter an error state thereby preventing potentially insecure operations (i.e., maintaining a secure state). The controller will reboot after a self-test failure. During reboot, memory is re-initialized, which wipes all keys and user data. If a self-test failure continues to occur, the controller will continue to reboot repeatedly and will require return to manufacturer.

141 The above tests are sufficient to demonstrate that the TSF is operating correctly by verifying the integrity of the TSF and the correct operation of cryptographic components.

### 6.7.6 FPT_TST_EXT.3

142 A software image stored in the image partition will be rejected by the ArubaOS Uboot bootloader if the image signature is invalid. This Firmware Integrity Test is performed at startup using RSA 2048-bit Signature Validation.

### 6.7.7 FPT_TUD_EXT.1

143 The TOE allows administrators to query the currently executing version of its firmware/software. For MC this can be accomplished by issuing the '*show version'* command, and the currently running and backup versions of firmware/software by issuing the '*show image version'* command. For RAP devices, the command '*show ap image version'* can be used to view the currently installed and running version of firmware/software. RAP devices can only have one version of firmware/software installed and running at any given time.

144 The TOE allows administrators to manually initiate firmware/software updates. Administrators can update the TOE executable code using image files manually downloaded from the Aruba support portal. The administrator may initiate an update from either the WebUI or CLI and can perform a system reboot post-installation for the update to take effect. Upgrade instructions are documented in the release notes for each software release, which will be posted in the same directory as the image file on the Aruba Support Portal.

145 Software images are verified at the time of receipt (before writing to the flash) and is also verified by the bootloader each time the TOE boots. The software images are hashed and cryptographically signed, and an image with an invalid signature will not be copied by the TOE into the image partition. RSA2048 and SHA256 are used to sign the image. The TOE computes a SHA256 sum over the entire software image. Using the RSA public key stored internally in the product, the computed hash is compared against the digital signature included in the software image to ensure it is authentic. Upon verification, the software image is written to the flash and the TOE automatically reboots. The 'Maintenance' tab in the WebUI contains the 'Software Management' menu which permits the user to display information about the currently installed version of the TOE software in addition to the current partitions.

146 When an update is initiated by the Controller, the TOE verifies the update via digital signature. Upon successful verification, MC will install the new image and the administrator has the option of rebooting immediately for the update to take effect or may elect to continue operation with the currently executing version (delayed activation). Should verification fail, the TOE will enter into an error state and the installation is aborted. The TOE's error state will allow direct console access only, where an administrator can change to a new file partition or TFTP a new image and re-boot.

147 The Remote Access Points must obtain TOE updates from the Controller. The administrator must first obtain the TOE update files from the portal as described above and load them onto the Controller. Upon connecting to the Aruba Mobility Controller, the TOE checks for a software update. If an update is available, the TOE initiates the software download. When the download completes, the TOE sends a message to the Aruba Mobility Controller, informing it that the TOE has either successfully downloaded the new software version, or that the preload has failed for some reason including digital signature verification failure. If the download fails, the TOE will retry the download after a brief waiting period. A software image that is downloaded from the Aruba Mobility Controller is both verified at the time of receipt (before writing to the flash) and is also verified by the bootloader each time the TOE boots. The software images are verified using digital signature as described above.

148 When a RAP connects to the controller, the controller checks the version of the RAP to ensure it is the same. IF the version do not match, the controller will force the RAP to update to the same version as the controller. Once the updated version is installed, the RAP will reboot immediately and the controller verifies that their versions now match. This ensures that TOE updates cannot lead to the situation where different TOE components are running different software versions.

### 6.7.8 FPT_STM_EXT.1

149 The TOE has an internal battery-backed hardware clock that provides reliable time stamps used for auditing. The internal clock can be set by the administrator and can be synchronized with a time signal obtained from an external NTP server. The clock is used to provide a timestamp for audit records, and to support timing elements of cryptographic functions, certificate validity checks, session timeouts, and unlocking of administrator accounts locked as a result of authentication failure. When an external NTP server is used, the TOE updates its system time using SHA1 authentication or IPsec to provide trusted communication between itself and the NTP time source.

## 6.8 TOE Access

### 6.8.1 FTA_SSL_EXT.1

150 Local inactive administrator sessions on the TOE are terminated after the configured timeout period. Session timeout can be configured for local CLI administrative sessions.

151 To define a timeout interval for a CLI session, use loginsession timeout <value> where value is from 1 to 3600 seconds.

### 6.8.2 FTA_SSL.3

152 Inactive remote administrator sessions on the TOE are terminated after the configured timeout period. Session timeout thresholds are configurable for the WebUI and CLI interfaces. To define a timeout interval for a WebUI session, use the command: web-server profile and session-timeout <session-timeout> where the session-timeout value can be any number of seconds from 30 to 3600, or by navigating in the WebUI to Configuration>System>Admin>Admin Authentication Options and setting the 'Idle session timeout' value. For the CLI, use loginsession timeout <value> where value is from 1 to 3600 seconds.

### 6.8.3 FTA_SSL.4

153 The TOE allows users to terminate their own local and remote CLI sessions by issuing the '*exit*' command. Terminating Web UI sessions requires the use of a "*Log out*" button in the user account menu.

### 6.8.4 FTA_TAB.1

154 The TOE displays an advisory warning banner regarding use of the TOE prior to establishing an administrator session. The administrator can configure the warning message displayed in the banner. The banner will be displayed when accessing the CLI locally via the console or remotely via SSH and when accessing the GUI. The configured message is identical across interfaces.

155 The TOE banner can be configured via CLI using the '*banner motd* =' command or via the GUI. On the MC GUI, navigate to 'Configuration > System'. Click on the 'Admin' tab and modify the 'Login banner text' field as necessary.

## 6.9 Trusted Path/Channels

### 6.9.1 FTP_ITC.1 & FTP_ITC.1/VPN

156 The TOE uses the IPsec/IKE protocol with certificates to establish VPN tunnels and to establish trusted channels between the Controller and the external authentication server, syslog server, and NTP server. These IPsec channels are peer-to-peer connections whereby the TOE can act as either the server or the client.

### 6.9.2 FTP_TRP.1/Admin

157 The TOE uses HTTPS/TLS to offer secure remote web GUI-based administration and SSH to offer a secure remote administration CLI. Each connection can be tunneled over IPsec for an additional layer of security. Administrators initiate a remote session that is secured (from disclosure and modification) using NIST-validated cryptographic operations, and all remote security management functions require the use of this secure channel.

# 7 Rationale

## 7.1 Conformance Claim Rationale

158 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is network device and VPN gateway consistent with the claimed Protection Profile and PP Module.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the CPP_ND_V2.2E, and MOD_VPNGW_V1.2.

c) **Security objectives.** As shown in section 18, the security objectives are reproduced directly from the CPP_ND_V2.2E, and MOD_VPNGW_V1.2.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the CPP_ND_V2.2E, and MOD_VPNGW_V1.2. Some SFR selections were chosen to ensure compliance with CSfC requirements for VPN gateways. No additional requirements have been specified.

## 7.2 Security Objectives Rationale

159 All security objectives are drawn directly from the claimed Protection Profiles and PP-Modules.

## 7.3 Security Requirements Rationale

160 All security requirements are drawn directly from the claimed PP and PP-Module in accordance with exact conformance. Refer to the Rationale sections of the PP and PP-Module as follows:

- CPP_ND_V2.2E – Appendix ‘E’
- MOD_VPNGW_V1.2 – Section 5.3 and Section 6.

## 7.4 SFR Distribution Between Components

161 The following table addresses the SFR distribution requirements between TOE components, as required in Section 3.4 of the NDcPP. For consistency, SFRs from MOD_VPNGW_V1.2 have been included to represent their distribution between components.

**Table 21: SFR Distribution Between Components**

| SFR | Dist. Requirement | Remote Access Points | Mobility Controller |
|---|---|---|---|
| FAU_GEN.1 | All | X | X |
| FAU_GEN_EXT.1 | All | X | X |
| FAU_GEN.1/VPN | n/a | | X |

| SFR | Dist. Requirement | Remote Access Points | Mobility Controller |
|---|---|---|---|
| FAU_GEN.2 | All | X | X |
| FAU_STG_EXT.1 | All | X | X |
| FAU_STG_EXT.4 | Feature Dependent | | X |
| FAU_STG_EXT.5 | Feature Dependent | X | |
| FCO_CPC_EXT.1 | All | X | X |
| FCS_CKM.1 | One | X | X |
| FCS_CKM.1/IKE | n/a | | X |
| FCS_CKM.2 | All | X | X |
| FCS_CKM.4 | All | X | X |
| FCS_COP.1/ DataEncryption | All | X | X |
| FCS_COP.1/ SigGen | All | X | X |
| FCS_COP.1/ Hash | All | X | X |
| FCS_COP.1/ KeyedHash | All | X | X |
| FCS_RBG_EXT.1 | All | X | X |
| FCS_HTTPS_EXT.1 | Feature Dependent | | X |
| FCS_IPSEC_EXT.1/ VPN | Feature Dependent | | X |
| FCS_IPSEC_EXT.1/ ITT | Feature Dependent | X | X |
| FCS_NTP_EXT.1 | Feature Dependent | | X |
| FCS_SSHS_EXT.1 | Feature Dependent | | X |
| FCS_TLSS_EXT.1 | Feature Dependent | | X |
| FIA_AFL.1 | One | | X |
| FIA_PMG_EXT.1 | One | | X |

| SFR | Dist. Requirement | Remote Access Points | Mobility Controller |
|---|---|---|---|
| FIA_UIA_EXT.1 | One | | X |
| FIA_UAU_EXT.2 | One | | X |
| FIA_UAU.7 | Feature Dependent | | X |
| FIA_X509_EXT.1/ Rev | Feature Dependent | | X |
| FIA_X509_EXT.1/ ITT | Feature Dependent | X | X |
| FIA_X509_EXT.2 | Feature Dependent | | X |
| FIA_X509_EXT.3 | Feature Dependent | X | X |
| FMT_MOF.1/ ManualUpdate | All | X | X |
| FMT_MOF.1/ Services | Feature Dependent | | X |
| FMT_MTD.1/ CoreData | All | X | X |
| FMT_MTD.1/ CryptoKeys | Feature Dependent | | X |
| FMT_SMF.1 | Feature Dependent | | X |
| FMT_SMF.1/VPN | n/a | | X |
| FMT_SMR.2 | One | | X |
| FPF_RUL_EXT.1 | n/a | | X |
| FPT_FLS.1/ SelfTest | n/a | | X |
| FPT_ITT.1 | Feature Dependent | X | X |
| FPT_ITT.1/Join | Feature Dependent | X | X |
| FPT_SKP_EXT.1 | All | X | X |
| FPT_APW_EXT.1 | Feature Dependent | | X |
| FPT_TST_EXT.1 | All | X | X |
| FPT_TST_EXT.3 | n/a | | X |

| SFR | Dist. Requirement | Remote Access Points | Mobility Controller |
|---|---|---|---|
| FPT_TUD_EXT.1 | All | X | X |
| FPT_STM_EXT.1 | All | X | X |
| FTA_SSL_EXT.1 | Feature Dependent | | X |
| FTA_SSL.3 | Feature Dependent | | X |
| FTA_SSL.4 | Feature Dependent | | X |
| FTA_TAB.1 | One | | X |
| FTP_ITC.1 | One | | X |
| FTP_ITC.1/ VPN | n/a | | X |
| FTP_TRP.1/ Admin | One | | X |