Aruba Mobility Controller with ArubaOS 8.6 Security Target (NDcPP21/STFFW13/VPNGW10)

Version 1.0
02/05/2021

Prepared for:

Aruba, a Hewlett Packard Enterprise Company

3333 Scott Blvd
Santa Clara, CA 95054

Prepared By:

Gossamer Laboratories

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Aruba Mobility Controller with ArubaOS 8.6 provided by Aruba, a Hewlett Packard Enterprise Company. The TOE is being evaluated as a network infrastructure device with VPN Gateway and Firewall capabilities.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – Aruba Mobility Controller with ArubaOS 8.6 Security Target (NDcPP21/STFFW13/VPNGW10)

ST Version – Version 1.0

ST Date – 02/05/2021

1.2 TOE Reference

TOE Identification – The TOE is Aruba Mobility Controller with ArubaOS version 8.6 and the following required software licenses:

- Policy Enforcement Firewall,
- RFprotect
- Advanced Cryptography

The TOE includes the following hardware and virtual appliance models:

Mobility Controller Hardware Appliances

Product Model	Part Number(s)	CPU
Aruba 9004 Mobility Controller	R1B25A	Intel Atom C3508 (Denverton)
Aruba 7005 Mobility Controller	JW636A	Broadcom XLP208 (MIPS64)
Aruba 7008 Mobility Controller	JX932A	Broadcom XLP208 (MIPS64)
Aruba 7010 Mobility Controller	JW703A	Broadcom XLP208 (MIPS64)
Aruba 7024 Mobility Controller	JW707A	Broadcom XLP208 (MIPS64)
Aruba 7030 Mobility Controller	JW711A	Broadcom XLP208 (MIPS64)
Aruba 7205 Mobility Controller	JW740A	Broadcom XLP316 (MIPS64)
Aruba 7210 Mobility Controller	JW746A	Broadcom XLP416 (MIPS64)
Aruba 7220 Mobility Controller	JW754A	Broadcom XLP432 (MIPS64)
Aruba 7240 Mobility Controller	JW762A	Broadcom XLP432 (MIPS64)
Aruba 7240XM Mobility Controller	JW830A	Broadcom XLP432 (MIPS64)
Aruba 7280 Mobility Controller	JX914A	Broadcom XLP780 (MIPS64)

Table 1 Mobility Controller Hardware Appliances

The table below shows the different model series based on maximum number of APs and users supported.

Product	Max. # of APs	Max. # of Users	Typical Deployment
Aruba 7000 Series	64	4,096	Branch Office/ Small Campus
Aruba 7200 Series	2,048	32,768	Headquarters / Large Campus
Aruba 9004	32	2,048	Branch Office / Small Campus

Table 2 Mobility Controller Deployments

Mobility Controller Virtual Appliances

- MC-VA-50
- MC-VA-250
- MC-VA-1k

The table below shows the different virtual models based on maximum number of APs and clients supported.

Aruba Mobility Controller Virtual Appliance	MC-VA-50	MC-VA-250	MC-VA-1K
Maximum AP Count	50 APs	250 APs	1,000 APs
Maximum Client Count	800	4,000	16,000

Table 3 MC Virtual Appliance Models and Capacities

VM Platforms
The Mobility Controller Virtual Appliances are deployed on ESXi version 6.5.0. The following virtual machine platforms are included in the evaluated configuration:

Name	CPU	Memory
HPE EdgeLine EL8000	Intel Xeon Gold 6212U (Cascade Lake)	128GB
Klas Telecom VoyagerVM3	Intel Xeon D (Coffee Lake)	96GB
IAS VPN Gateway Module Classic Plus	Intel Core i7 (Skylake)	32GB
Pacstar 451/3	Intel Xeon D (Coffee Lake)	32GB
Pacstar 451/3	Intel Xeon E3 (Coffee Lake)	32GB
DTECH M3-SE-SVR4	Intel Xeon E3 v5-1505L (Skylake)	32GB
DTECH M3x	Intel Core i5 (Skylake)	32GB
GTS NXGEN-L11/12	Intel Core i7 (Coffee Lake)	16GB

Table 4 VM Platforms

TOE Developer – Aruba, a Hewlett Packard Enterprise Company

Evaluation Sponsor – Aruba, a Hewlett Packard Enterprise Company

1.3 TOE Overview

The Target of Evaluation (TOE) is Aruba Mobility Controller with ArubaOS 8.6. The TOE is a multi-purpose network device that includes stateful traffic filter firewall and VPN gateway capabilities.

1.4 TOE Description

The Aruba Mobility Controller platform serves as a gateway between wired and wireless networks and provides command and control over Aruba Access Points (APs) within an Aruba dependent wireless network.

The Aruba Mobility Controllers (MCs) and Aruba Virtual Mobility Controllers (VMCs) are wireless switch hardware and virtual appliances that provide a wide range of security services and features including wireless and wired network mobility, security, centralized management, auditing, authentication, secure remote access, self-verification of integrity and operation, stateful traffic filtering and VPN gateway functionality.

The ArubaOS is a suite of mobility applications that runs on all Aruba controllers and allows administrators to configure and manage the wireless and mobile user environment. The TOE is generally deployed in a configuration consisting of one or more Aruba mobility controllers (MC and/or VMC) and multiple Aruba wireless APs[1]. A simple TOE deployment is depicted in Figure 1 below. Note that the Mobility Controller in the figure represents an MC or a VMC.

[1] Aruba Access Points are part of the operating environment of the TOE.

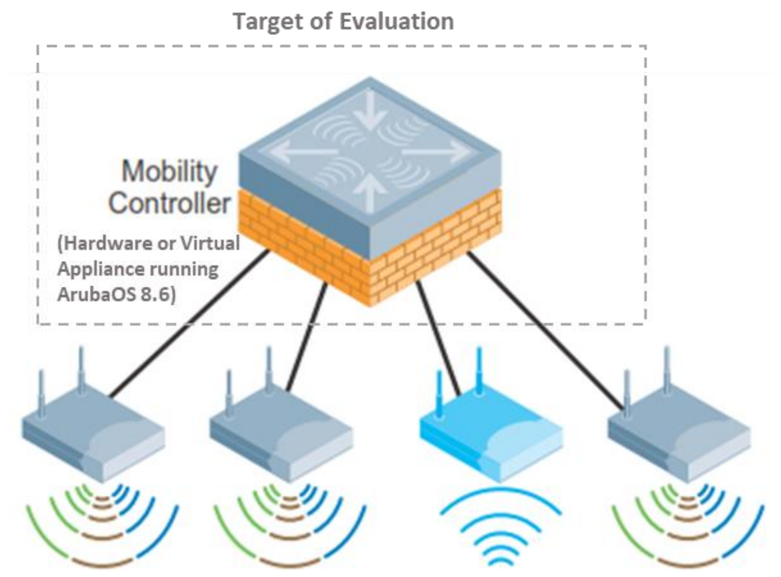

Figure 1: Simple TOE Usage Scenario

The TOE performs stateful packet filtering on network packets processed by the TOE. Filtering rules may be applied to appliance Ethernet interfaces and to user roles (for wireless clients as described above) to allow fine grained control over network traffic.

As a VPN gateway – a device at the edge of a private network that terminates an IPsec tunnel – the TOE provides device authentication, confidentiality, and integrity of information traversing a public or untrusted network. The TOE provides packet filtering and secure IPsec tunneling. This functionality may be used with VPN clients or with other VPN gateways (i.e. site-to-site VPN).

1.4.1 TOE Architecture

Aruba Mobility Controllers (MCs) are hardware appliances consisting of a multicore network processor, Ethernet interfaces, and required supporting circuitry and power supplies enclosed in a metal chassis. Aruba Virtual Mobility Controllers (VMCs) consist of a 64-bit virtualized software-based managed platform on virtual machine (VM) architecture. The Aruba VMC operates on x86 platforms in a hypervisor environment.

The ArubaOS software running on the MCs and VMCs consists of two main components:

- Control Plane (CP) – implements functions which can be handled at lower speeds such as Mobility Controller system management (CLI and Web GUI), VMC system management (CLI and Web GUI), user authentication (e.g. RADIUS), Internet Key Exchange (IKE), auditing/logging (syslog), Wireless IDS (WIDS), and termination of protocols operating at the system level (e.g. SSH, TLS, NTP, etc.). The Control Plane runs the Linux operating system along with various user-space applications (described below).
- Data Plane (DP) - implements functions that must be handled at high speeds such as high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging), termination of 802.11 associations/sessions, tunnel termination (IPsec), stateful firewall and deep packet inspection functions, and cryptographic acceleration. The Data Plane runs a lightweight, proprietary real-time OS which is known as "SOS" (an acronym which used to mean "SiByte Operating System" for an earlier generation of Mobility Controller that used the SiByte CPU). On the Mobility Controller hardware appliances, SOS runs on separate CPU cores. On the Virtual Mobility Controller appliances, SOS is a process running under Linux.

The Control Plane and Data Plane are inseparable. Administrators install the MC software by loading a single file, identified as "ArubaOS". Administrators install the VMC by creating a virtual machine in the ESXi client interface and then applying the VMC OVF template to the virtual machine, identified as "ArubaOS". Internally, the controllers unpack the ArubaOS software image into its various components. A given ArubaOS software image has a single version number and includes all software components necessary to operate the MC and VMC appliances as well as the APs which are in the operating environment of the TOE.

The CP runs the Linux OS, along with various custom user-space applications which provide the following CP functions:

- Monitors and manages critical system resources, including processes, memory, and flash
- Manages system configuration and licensing
- Manages an internal database used to store licenses, user authentication information, etc
- Provides network anomaly detection, hardware monitoring, mobility management, wireless management, and radio frequency management services
- Provides a Command Line Interface (CLI)
- Provides a web-based (HTTPS/TLS) management UI for the MCs and VMCs
- Provides authentication services for the system management interfaces (CLI, web GUI)
- Provides IPsec key management services for VPN users, and connections with other Aruba mobility controllers
- Provides network time protocol service, point to point tunneling protocol services for users, layer 2 tunneling protocol services for users, SSH services for incoming management connections, SNMP client/agent services, and protocol independent multicast (routing) services for the controller
- Provides syslog services by sending logs to the operating environment

The Linux OS running on the CP is a version 2.6.32 kernel for the MCs and a version 3.18.26 kernel for the VMCs. Linux is a soft real-time, multi-threaded operating system that supports memory protection between processes. Only Aruba provided interfaces are used, and the CLI is a restricted command set. Administrators do not have access to the Linux command shell or operating system.

All Aruba Mobility Controller and Virtual Mobility Controller models run the same ArubaOS 8.6 software and include the same ArubaOS Crypto Module. Regardless of the different hardware and virtual platforms, the security functionality remains the same. The differences in the platforms are in the processing speed, throughput, memory capacity, storage, physical interfaces, number of ports, etc., and are based on performance and scalability requirements. All models run the same code with the only differences being the hardware specific code for the differently scaled hardware and the virtual nature of the hardware ports on the VMs.

The Virtual Mobility Controller uses gigabit Ethernet interfaces just as a hardware-based mobility controller does, but these interfaces map to virtual interfaces created in the underlying hypervisor. Those interfaces, in turn, may map directly to physical ports. The device drivers on all VM platforms are identical because the ArubaOS is being run on a hypervisor. Within the hypervisor, there may be slight differences in device drivers (mostly for network interfaces), however, the device drivers are not used to enforce any TOE security functions.

Although the TOE models have different specifications (in terms of performance and scalability), they all provide the same security functions described in the ST; therefore, they have been considered to be the same for the purposes of the ST description.

1.4.1.1 Physical Boundaries

The TOE consists of the following components:

- Aruba Mobility Controllers: 7280, 9004, 7005, 7008, 7010, 7024, 7030, 7205, 7210, 7220, 7240, 7240XM

- Aruba Virtual Mobility Controllers: MC-VA-50, MC-VA-250, MC-VA-1k
- ArubaOS version 8.6

These components are identified and described in section 1.2 and section 1.4.1 of this ST.

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

Required Software Module	Description
Policy Enforcement Firewall Next Generation	Provides identity-based security for wired and wireless clients. Stateful firewall enables classification based on client identity, device type, location, and time of day, and provides differentiated access for different classes of users.
RFprotect	Detects, classifies and limits designated wireless security threats such as rogue APs, DoS attacks, malicious wireless attacks, impersonations, and unauthorized intrusions. Eliminates need for separate system of RF sensors and security appliances. Also provides spectrum intelligence and spectrum visibility when used with compatible AP platforms.
Advanced Cryptography	Required for SuiteB, AES-GCM and ECDSA functionality.

Table 5 Required Licenses

The TOE operates with the following components in the Operating Environment:

- One or more of the following Aruba Access Points running ArubaOS 8.6:

Product Model	Part Number(s)
AP-203R Access Point	JY715A
AP-203RP Access Point	JY723A
AP-204 Access Point	JW163A
AP-205 Access Point	JW165A
AP-205H Access Point	JW167A
AP-214 Access Point	JW169A
AP-215 Access Point	JW798A
AP-224 Access Point	JW173A
AP-225 Access Point	JW175A
AP-228 Access Point	JW183A
AP-274 Access Point	JW177A
AP-275 Access Point	JW179A
AP-277 Access Point	JW181A
AP-303H Access Point	JW681A
AP-304 Access Point	JX937A
AP-305 Access Point	JX938A

AP-314 Access Point	JW796A
AP-315 Access Point	JW798A
AP-324 Access Point	JW185A
AP-325 Access Point	JW187A
AP-334 Access Point	JW800A
AP-335 Access Point	JW802A
AP-504 Access Point	R2H34A
AP-505 Access Point	R2H39A
AP-514 Access Point	Q9H67A
AP-515 Access Point	Q9H73A
AP-534 Access Point	JZ342A
AP-535 Access Point	JZ347A
AP-555 Access Point	JZ367A

Table 6 Aruba Access Points

- Audit Server – The TOE utilizes an external syslog server to store audit records.
- Authentication Server – The TOE utilizes RADIUS and TACACs+ servers to authenticate users.
- Time Server – The TOE uses a Network Time Protocol (NTP) server to synchronize its system clock with a central time source.
- Web Browser – The remote administrator uses a web browser to access the Web GUI interface.
- SSH Client – The remote administrator uses an SSH client to access the CLI.
- VPN Gateway peers/VPN Clients - When acting as a VPN gateway, the TOE may communicate with other VPN gateways or with VPN clients.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Security audit
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The TOE is designed to be able to generate logs for a wide range of security relevant events including start-up and shutdown of the TOE, all administrator actions, and all events identified in Table 8 Auditable Events. The TOE can be configured to store the logs locally so they can be accessed by an administrator or alternately to send the logs to a designated syslog server in the operational environment.

1.4.1.2.2 Cryptographic support

The TOE includes cryptographic modules that provide key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher-level cryptographic protocols including IPsec, SSH, and TLS/HTTPS.

1.4.1.2.3 User data protection

The TOE ensures that any data packets passing through do not inadvertently contain any residual information that might be disclosed inappropriately.

1.4.1.2.4 Firewall

The TOE performs stateful packet filtering. Filtering rules may be applied to appliance Ethernet interfaces or to user-roles (wireless clients connecting through APs are placed into user-roles). Stateful packet filter policies are applied to user-roles to allow fine grained control over wireless traffic.

1.4.1.2.5 Identification and authentication

The TOE requires administrators to be identified and authenticated before they can access any TOE security functions. The TOE supports role-based authentication, so user accounts are assigned predefined roles which restrict them based on their assigned role. The TOE maintains these administrator and user attributes which can be defined locally with user names and passwords or can be defined in the context of RADIUS or TACACS+ services. Authentication can be either locally or remotely through an external authentication server, or internally. After an administrator-specified number of failed attempts, the user account is locked out. The TOE's password mechanism provides configuration for a minimum password length. The TOE also protects, stores and allows authorized administrators to load X.509.v3 certificates for use to support authentication for IPsec connections.

1.4.1.2.6 Security Management

The TOE provides the administrator role 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. The role must have the appropriate access privileges or access will be denied. The TOE's cryptographic functions ensure that only secure values are accepted for security attributes.

1.4.1.2.7 Packet Filtering

The TOE may be used 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 and secure IPsec tunneling. The tunnels can be established between two trusted VPN peers as well as between remote VPN clients and the TOE. An administrator can configure security policies that determine whether to block, allow, or log a session based on traffic attributes such as the source and destination security zone, the source and destination IP address, the application, user, and the service.

1.4.1.2.8 Protection of the TSF

The TOE has an internal hardware clock that provides reliable time stamps used for auditing. The internal clock may be synchronized with a time signal obtained from an external trusted NTP server. The TOE stores passwords on flash using a SHA1 hash and does not provide any interfaces that allow passwords or keys to be read.

The TOE runs self-tests during power up and periodically during operation to ensure the correct operation of the cryptographic functions and TSF hardware. There is an option for the administrator to verify the integrity of stored TSF executable code.

The TOE includes mechanisms so that the administrator can determine the TOE version and update the TOE securely using digital signatures.

1.4.1.2.9 TOE access

The TOE allows administrators to configure a period of inactivity for administrator sessions. Once that time period has been reached while the session has no activity, the session is terminated. All users may also terminate their own sessions at any time. A warning banner is displayed at the management interfaces (Web GUI and CLI) to advise users on appropriate use and penalty for misuse of system.

In order to limit access to the administrative functions, the TOE can be configured to deny remote VPN clients based on the time/date, IP address (location), as well as information retained in a blacklist. The TOE assigns a private IP address (internal to the trusted network for which the TOE is the headend) to a VPN client upon successful establishment of a session.

1.4.1.2.10 Trusted path/channels

The TOE uses IPsec to provide an encrypted channel between itself and third-party trusted IT entities in the operating environment including external syslog server, external authentication server, NTP server and VPN Gateway/Client.

The TOE secures remote communication with administrators by implementing TLS/HTTPS for remote Web UI access and SSHv2 for CLI access. In each case, both the integrity and disclosure protection is ensured via the secure protocol. If the negotiation of a secure session fails or if the user cannot be authenticated for remote administration, the attempted session will not be established.

1.4.2 TOE Documentation

The TOE includes the following guidance documents:

- Aruba OS 8.6 Supplemental Guidance (Common Criteria Configuration Guidance), Version 1.10, February 2021

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, Version 1.0, 2020-03-06

      The PP-Configuration includes the following components:
        - collaborative Protection Profile for Network Devices, Version 2.1, 24 September 2018 (NDcPP21)
        - PP-Module for Stateful Traffic Filter Firewalls, Version 1.3, 27 September 2019 (STFFW13)
        - PP-Module for Virtual Private Network (VPN) Gateways, Version 1.0, 2019-09-17 (VPNGW10)
- Technical Decisions:

TD #	TD Name	Protection Profile	Applied to this TOE
TD0549	Consistency of Security Problem Definition update for MOD_VPNGW_v1.0 and MOD_VPNGW_v1.1	VPNGW10	Section 6.1.2 of VPNGW10
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	NDcPP21	AVA_VAN.1
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	STFFW13	FFW_RUL_EXT.1.8
TD0538	NIT Technical Decision for Outdated link to allowed-with list	NDcPP21	Section 2 of the PP
TD0536	NIT Technical Decision for Update Verification Inconsistency	NDcPP21	AGD_OPE.1
TD0535	NIT Technical Decision for Clarification about digital signature algorithms for FPT_TUD.1	NDcPP21	Application Note 32 in NDcPP21 clarification
TD0534	NIT Technical Decision for Firewall IPv4 & IPv6 testing by default	STFFW13	FFW_RUL_EXT.1
TD0533	NIT Technical Decision for FTP_ITC.1 with signed downloads	NDcPP21	FTP_ITC.1
TD0532	NIT Technical Decision for Use of seeds with higher entropy	NDcPP21	FCS_RBG_EXT.1.2
TD0531	NIT Technical Decision for Challenge-Response for Authentication	NDcPP21	FCS_SSHS_EXT.1

TD0530	NIT Technical Decision for FCS_TLSC_EXT.1.1 5e test clarification	NDcPP21	Not applied. FCS_TLSC_EXT.1.1 is not claimed.
TD0529	NIT Technical Decision for OCSP and Authority Information Access extension	NDcPP21	FIA_X509_EXT.1/Rev , FIA_X509_EXT.2
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	NDcPP21	FCS_NTP_EXT.1.4
TD0520	VPN Gateway SFR Rationale	VPNGW10	Section 5.4 of VPNGW10
TD0511	VPN GW Conformance claim to allow for a PP-Module	VPNGW10	Section 2, Package Claims
TD0484	NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3	NDcPP21	FTA_SSL_EXT.1, FTA_SSL.3
TD0483	NIT Technical Decision for Applicability of FPT_APW_EXT.1	NDcPP21	FPT_APW_EXT.1
TD0482	NIT Technical Decision for Identification of usage of cryptographic schemes	NDcPP21	FCS_CKM.1, FCS_CKM.2, ND SD V2.1
TD0481	NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers	NDcPP21	Not applied. FCS_TLSC_EXT.x.2 and FCS_DTLSC_EXT.x.2 are not claimed.
TD0480	NIT Technical Decision for Granularity of audit events	NDcPP21	FAU_GEN.1, ND SD V2.1
TD0478	NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	NDcPP21	FIA_X509_EXT.1
TD0477	NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update	NDcPP21	FPT_TUD_EXT.1, ND SD V2.1
TD0475	NIT Technical Decision for Separate traffic consideration for SSH rekey	NDcPP21	FCS_SSHS_EXT.1.8, ND SD V2.1
TD0453	NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH servers	NDcPP21	Not applied. FCS_SSHC_EXT.1 is not claimed.
TD0451	NIT Technical Decision for ITT Comm UUID Reference Identifier	NDcPP21	FCS_TLSS_EXT.1.2
TD0450	NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message	NDcPP21	FCS_TLSS_EXT.*.3, ND SD v2.1
TD0447	NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	NDcPP21	FCS_SSHS_EXT.1.7
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA	NDcPP21	FTA_SSL.3, ND SD V2.1
TD0424	NIT Technical Decision for NDcPP v2.1 Clarification - FCS_SSHC/S_EXT1.5	NDcPP21	FCS_SSHS_EXT.1.5
TD0423	NIT Technical Decision for Clarification about application of RfI#201726rev2	NDcPP21	ND SD V2.1
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	NDcPP21	FCS_SSHS_EXT.1, ND SD V2.1

TD0411	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused	NDcPP21	Not applied. FCS_SSHC_EXT.1 is not claimed.
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	NDcPP21	FAU_GEN.1, ND SD V2.1
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	NDcPP21	FIA_AFL.1, ND SD v2.1
TD0408	NIT Technical Decision for local vs. remote administrator accounts	NDcPP21	FIA_AFL.1, FIA_UAU_EXT.2, FMT_SMF.1
TD0407	NIT Technical Decision for handling Certification of Cloud Deployments	NDcPP21	Not applied. This ST does not include Cloud Deployments.
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	NDcPP21	FCS_CKM.2, ND SD V2.1
TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs	NDcPP21	FTP_ITC.1
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	NDcPP21	FCS_CKM.1, FCS_CKM.2
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	NDcPP21	FIA_X509_EXT.2, ND SD V2.1
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	NDcPP21	FCS_SSHS_EXT.1
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests	NDcPP21	FCS_COP.1/DataEncryption, ND SD V2.1
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	NDcPP21	Not applied. FCS_TLSC_EXT.1 not claimed.
TD0395	NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2	NDcPP21	Not applied. FCS_TLSS_EXT.2 not claimed.

2.1 Conformance Rationale

The ST conforms to the NDcPP21/STFFW13/VPNGW10. The security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the NDcPP21/STFFW13/VPNGW10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP21/STFFW13/VPNGW10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP21/STFFW13/VPNGW10 should be consulted if there is interest in that material.

In general, the NDcPP21/STFFW13/VPNGW10 has defined Security Objectives appropriate for Network Devices with VPN Gateway, and Firewall capabilities and as such are applicable to the TOE.

3.1 Security Objectives for the Operational Environment

OE.CONNECTIONS: 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. (TD0356 applied)

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.CONNECTIONS (NDcPP21) TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

OE.CONNECTIONS (VPNGW10) 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.

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.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

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.

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner. 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.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP21/STFFW13/VPNGW10.

The NDcPP21/STFFW13/VPNGW10 defines the following extended requirements and since they are not redefined in this ST, the NDcPP21/STFFW13/VPNGW10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP21:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
- VPNGW10:FCS_IPSEC_EXT.1: IPsec Protocol
- NDcPP21:FCS_NTP_EXT.1: NTP Protocol
- NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
- NDcPP21:FCS_TLSS_EXT.1: TLS Server Protocol
- STFFW13: FFW_RUL_EXT.1: Stateful Traffic Filtering
- NDcPP21:FIA_PMG_EXT.1: Password Management
- VPNGW10:FIA_PSK_EXT.1: Pre-Shared Key Composition
- NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP21:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW10:FPF_RUL_EXT.1: Rules for Packet Filtering
- NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP21:FPT_TST_EXT.1: TSF testing
- VPNGW10:FPT_TST_EXT.3: Extended: TSF Testing
- NDcPP21:FPT_TUD_EXT.1: Trusted update
- VPNGW10:FPT_TUD_EXT.1: Extended: TSF Testing
- NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking
- VPNGW10:FTA_VCM_EXT.1: VPN Client Management

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDcPP21/STFFW13/VPNGW10. The refinements and operations already performed in the NDcPP21STFFW13/VPNGW10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP21/STFFW13/VPNGW10 and any residual operations have been completed herein. Of particular note, the NDcPP21/STFFW13/VPNGW10 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP21/STFFW13/VPNGW10 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP21/STFFW13/VPNGW10 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP21/STFFW13/VPNGW10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Aruba Mobility Controller TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP21:FAU_GEN.1: Audit Data Generation
	NDcPP21:FAU_GEN.2: User identity association
	NDcPP21:FAU_STG.1: Protected audit trail storage
	NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP21:FCS_CKM.1: Cryptographic Key Generation
	VPNGW10:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication)
	NDcPP21:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP21:FCS_CKM.4: Cryptographic Key Destruction
	VPNGW10:FCS_COP.1/DataEncryption: Cryptographic Operation (Data Encryption/Decryption)
	NDcPP21:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP21:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP21:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP21:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP21:FCS_HTTPS_EXT.1: HTTPS Protocol
	NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
	VPNGW10:FCS_IPSEC_EXT.1: IPsec Protocol
	NDcPP21:FCS_NTP_EXT.1: NTP Protocol
	NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
	NDcPP21:FCS_TLSS_EXT.1: TLS Server Protocol
FDP: User Data Protection	STFFW13:FDP_RIP.2: Full Residual Information Protection
FFW: Stateful Traffic Filtering	STFFW13:FFW_RUL_EXT.1: Stateful Traffic Filtering
FIA: Identification and authentication	NDcPP21:FIA_AFL.1: Authentication Failure Management
	NDcPP21:FIA_PMG_EXT.1: Password Management

	VPNGW10:FIA_PSK_EXT.1: Pre-Shared Key Composition
	NDcPP21:FIA_UAU.7: Protected Authentication Feedback
	NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP21/VPNGW10:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP21:FMT_MOF.1/Functions: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/Services: Management of security functions behaviour
	NDcPP21:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP21:FMT_MTD.1/CryptoKeys: Management of TSF data
	VPNGW10:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP21:FMT_SMF.1: Specification of Management Functions
	STFFW13:FMT_SMF.1/FFW Specification of Management Functions
	VPNGW10:FMT_SMF.1: Specification of Management Functions
	NDcPP21:FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	VPNGW10:FPF_RUL_EXT.1: Rules for Packet Filtering
FPT: Protection of the TSF	NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
	VPNGW10:FPT_FLS.1/SelfTest: Fail Secure (Self-test Failures) - Self Test
	NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
	NDcPP21:FPT_TST_EXT.1: TSF testing
	VPNGW10:FPT_TST_EXT.1:TSF testing
	VPNGW10:FPT_TST_EXT.3: Self-Test with Defined Methods
	NDcPP21:FPT_TUD_EXT.1: Trusted update
	VPNGW10:FPT_TUD_EXT.1: Extended: TSF Testing
FTA: TOE access	NDcPP21:FTA_SSL.3: TSF-initiated Termination
	NDcPP21:FTA_SSL.4: User-initiated Termination
	NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP21:FTA_TAB.1: Default TOE Access Banners
	VPNGW10:FTA_TSE.1: TOE Session Establishment
	VPNGW10:FTA_VCM_EXT.1: VPN Client Management
FTP: Trusted path/channels	NDcPP21:FTP_ITC.1: Inter-TSF trusted channel
	VPNGW10:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications)
	NDcPP21:FTP_TRP.1/Admin: Trusted Path

Table 7 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP21/STFFW13/VPNGW10: FAU_GEN.1)

NDcPP21/STFFW13/VPNGW10:FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions 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 8.

Requirement	Auditable Events	Additional Content
NDcPP21:FAU_GEN.1	None	None
NDcPP21:FAU_GEN.2	None	None
NDcPP21:FAU_STG.1	None	None
NDcPP21:FAU_STG_EXT.1	None	None
NDcPP21:FCS_CKM.1	None	None
VPNGW10:FCS_CKM.1/IKE	None	None
NDcPP21:FCS_CKM.2	None	None
NDcPP21:FCS_CKM.4	None	None
NDcPP21/VPNGW10:FCS_COP.1/DataEncryption	None	None
NDcPP21:FCS_COP.1/Hash	None	None
NDcPP21:FCS_COP.1/KeyedHash	None	None
NDcPP21:FCS_COP.1/SigGen	None	None
NDcPP21:FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
NDcPP21:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
VPNGW10:FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
NDcPP21:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity of new/removed time server
NDcPP21:FCS_RBG_EXT.1	None	None
NDcPP21:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP21:FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
STFFW13:FDP_RIP.2	None	None
STFFW13:FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol

		TOE Interface
NDcPP21:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_PMG_EXT.1	None	None
VPNGW10:FIA_PSK_EXT.1	None	None
NDcPP21:FIA_UAU.7	None	None
NDcPP21:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
NDcPP21/VPNGW10:FIA_X509_EXT.2	None	None
NDcPP21/VPNGW10:FIA_X509_EXT.3	None	None
NDcPP21:FMT_MOF.1/Functions	None	None
NDcPP21:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP21:FMT_MOF.1/Services	Starting and stopping of services.	None
NDcPP21:FMT_MTD.1/CoreData	None	None
NDcPP21:FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None
VPNGW10:FMT_MTD.1/CryptoKeys	None	None
NDcPP21:FMT_SMF.1	All management activities of TSF data.	
STFFW13:FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	None
VPNGW10:FMT_SMF.1	None	None
NDcPP21:FMT_SMR.2	None	None
VPNGW10:FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol
NDcPP21:FPT_APW_EXT.1	None	None
VPNGW10:FPT_FLS.1/SelfTest	None	None
NDcPP21:FPT_SKP_EXT.1	None	None
NDcPP21:FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time

	to time need to be logged. See also application note on FPT_STM_EXT.1)	for success and failure (e.g., IP address).
NDcPP21/VPNGW10:FPT_TST_EXT.1	None	None
VPNGW10:FPT_TST_EXT.3	None	None
NDcPP21:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
VPNGW10:FPT_TUD_EXT.1	None	None
NDcPP21:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
NDcPP21:FTA_SSL.4	The termination of an interactive session.	None
NDcPP21:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	None
NDcPP21:FTA_TAB.1	None	None
VPNGW10:FTA_TSE.1	None	None
VPNGW10:FTA_VCM_EXT.1	None	None
NDcPP21:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
VPNGW10:FTP_ITC.1/VPN	None	None
NDcPP21:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None

Table 8 Auditable Events

NDcPP21/STFFW13/VPNGW10: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 8.

ST Application Note: *The FAU_GEN.1 requirement is only found in the NDcPP21 base PP. This iteration of FAU_GEN.1 extends the FAU_GEN.1 requirement from NDcPP21 with the additional audit events identified in the VPNGW10, and STFFW13.*

5.1.1.2 User identity association (NDcPP21:FAU_GEN.2)

NDcPP21: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.

5.1.1.3 Protected audit trail storage (NDcPP21:FAU_STG.1)

NDcPP21:FAU_STG.1.1
The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
NDcPP21:FAU_STG.1.2
The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.1.1.4 Protected Audit Event Storage (NDcPP21:FAU_STG_EXT.1)

NDcPP21: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.
NDcPP21:FAU_STG_EXT.1.2
The TSF shall be able to store generated audit data on the TOE itself.
[

- ***TOE shall consist of a single standalone component that stores audit data locally***

]
NDcPP21:FAU_STG_EXT.1.3
The TSF shall [***overwrite previous audit records according to the following rule: [FIFO – First in, First out]***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP21:FCS_CKM.1)

NDcPP21: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' [selection: P-256, P-384] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4,***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1,***
- ***FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3***].

5.1.2.2 Cryptographic Key Generation (for IKE Peer Authentication)- IKE (VPNGW10:FCS_CKM.1/IKE))

VPNGW10: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-256, P-384 and [no other curves]***]

and [***no other key generation algorithms***]

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

5.1.2.3 Cryptographic Key Establishment (NDcPP21:FCS_CKM.2)

NDcPP21:FCS_CKM.2.1
The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***"RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"*** (TD0402 applied)
- ***Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',***
- ***Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',***
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3***].

5.1.2.4 Cryptographic Key Destruction (NDcPP21:FCS_CKM.4)

NDcPP21: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-pass] overwrite consisting of [zeroes]***]

that meets the following: No Standard.

5.1.2.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP21/VPNGW10:FCS_COP.1/DataEncryption)

NDcPP21/VPNGW10:FCS_COP.1.1/DataEncryption
The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC(IPsec and TLS), GCM (IPsec and TLS)***] and [***CTR(SSH only)***] mode and cryptographic key sizes [***128 bits(IPsec, TLS, SSH), 192 bits (IPsec only), 256 bits(IPsec, TLS, SSH)***] 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*]**.

VPNGW10 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.*

***ST Application Note**: In the TOE, GCM and CBC are supported for TLS and IPsec with 128 bit and 256 bit keys and CBC is also supported for IPsec with 192 bits. CTR is supported for SSH with 128 bit and 256 bit keys.*

5.1.2.6 Cryptographic Operation (Hash Algorithm) (NDcPP21:FCS_COP.1/Hash)

NDcPP21: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***] and message digest sizes [***160, 256, 384***] bits that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP21:FCS_COP.1/KeyedHash)

NDcPP21: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 bit, 256 bit, 384 bit**] and message digest sizes [***160, 256, 384***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

5.1.2.8 Cryptographic Operation (Signature Generation and Verification) (NDcPP21:FCS_COP.1/SigGen)

NDcPP21: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]***]

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***].

5.1.2.9 HTTPS Protocol (NDcPP21:FCS_HTTPS_EXT.1)

NDcPP21:FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC 2818.
NDcPP21:FCS_HTTPS_EXT.1.2
The TSF shall implement HTTPS using TLS.
NDcPP21: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.

5.1.2.10 IPsec Protocol (NDcPP21/VPNGW10:FCS_IPSEC_EXT.1)

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.1
The TSF shall implement the IPsec architecture as specified in RFC 4301.
NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.2
The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.
NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.3
The TSF shall implement [***transport mode, tunnel mode***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128, AES-CBC-192, 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***]

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [

- ***IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence numbers], and [RFC 4868 for hash functions],***
- ***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]***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [***IKEv1, IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602)***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [

- ***IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured within [1-24] hours],***
- ***IKEv2 SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured within [1-24] hours]***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.8

The TSF shall ensure that [

- ***IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on[***
  - ***length of time, where the time values can be configured within [1-8] hours],***
- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured within [1-8] hours]***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.9

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.

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv1, IKEv2***] exchanges of length [***at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.11

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

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.12

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

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.13

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 [***Pre-shared Keys***].

NDcPP21/VPNGW10:FCS_IPSEC_EXT.1.14

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***].

VPNGW10 Application Note: *FCS_IPSEC_EXT.1.4: 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. When an AES-CBC algorithm is selected, at least one SHA-based HMAC must also be chosen. If only an AES-GCM algorithm is selected, then a SHA-based HMAC is not required since AES-GCM satisfies both confidentiality and integrity functions.*

VPNGW10 Application Note: *FCS_IPSEC_EXT.1.11: 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, and by adding DH group 15 as a new selection. 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.*

VPNGW10 Application Note: *FCS_IPSEC_EXT.1.14: 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.*

5.1.2.11 NTP Protocol (NDcPP21:FCS_NTP_EXT.1)

NDcPP21:FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [***NTP v4 (RFC 5905)***].

NDcPP21:FCS_NTP_EXT.1.2
The TSF shall update its system time using [***[IPsec] to provide trusted communication between itself and an NTP time source.***].

NDcPP21:FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses

NDcPP21:FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources.

5.1.2.12 Random Bit Generation (NDcPP21:FCS_RBG_EXT.1)

NDcPP21: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)***].

NDcPP21:FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[2] software-based noise source, [2] hardware-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.

5.1.2.13 SSH Server Protocol (NDcPP21:FCS_SSHS_EXT.1)

NDcPP21:FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344***]. (TD0398 applied)

NDcPP21:FCS_SSHS_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***password-based***].

NDcPP21:FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than [**256k**] bytes in an SSH transport connection are dropped.

NDcPP21: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***].

NDcPP21:FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa***] as its public key algorithm(s) and rejects all other public key algorithms. (TD0424 applied)

NDcPP21:FCS_SSHS_EXT.1.6

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

NDcPP21:FCS_SSHS_EXT.1.7

The TSF shall ensure that [***diffie-hellman-group14-sha1***] and [***no other methods***] are the only allowed key exchange methods used for the SSH protocol.

NDcPP21: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. (TD0475 applied)

5.1.2.14 TLS Server Protocol (NDcPP21:FCS_TLSS_EXT.1)

NDcPP21: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
].

NDcPP21:FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***TLS 1.1,***].

NDcPP21:FCS_TLSS_EXT.1.3

The TSF shall [***perform RSA key establishment with key size [2048 bits], generate EC Diffie-Hellman parameters over NIST curves [secp256r1] and no other curves***].

5.1.3 User data protection (FDP)

5.1.3.1 Full Residual Information Protection (STFFW13:FDP_RIP.2)

STFFW13:FDP_RIP.2.1

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

5.1.4 Stateful Traffic Filtering (FFW)

5.1.4.1 Stateful Traffic Filtering (STFFW13:FFW_RUL_EXT.1)

STFFW13:FFW_RUL_EXT.1.1

The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

STFFW13:FFW_RUL_EXT.1.2

The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

ICMPv4
- o Type
- o Code

ICMPv6
- o Type
- o Code

IPv4
- o Source address
- o Destination Address
- o Transport Layer Protocol

IPv6
- o Source address
- o Destination Address
- o Transport Layer Protocol
- o [***IPv6 Extension header type [no other field]***]

TCP
- o Source Port
- o Destination Port

UDP
- o Source Port
- o Destination Port

and distinct interface.

STFFW13:FFW_RUL_EXT.1.3
The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit or drop with the capability to log the operation.

STFFW13:FFW_RUL_EXT.1.4
The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

STFFW13:FFW_RUL_EXT.1.5
The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [***ICMP***] based on the following network packet attributes:

1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
2. UDP: source and destination addresses, source and destination ports;
3. [***ICMP: source and destination addresses,***]

b) Remove existing traffic flows from the set of established traffic flows based on the following: [***session inactivity timeout, completion of the expected information flow***]

STFFW13:FFW_RUL_EXT.1.6
The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

a) The TSF shall drop and be capable of [***logging***] packets which are invalid fragments;
b) The TSF shall drop and be capable of [***logging***] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address 'reserved for future use' (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
i) [***no other rules***]

STFFW13:FFW_RUL_EXT.1.7
The TSF shall be capable of dropping and logging according to the following rules:
a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

STFFW13:FFW_RUL_EXT.1.8
The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

STFFW13:FFW_RUL_EXT.1.9
The TSF shall deny packet flow if a matching rule is not identified.

STFFW13:FFW_RUL_EXT.1.10
The TSF shall be capable of limiting an administratively defined number of half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [***counted***]

5.1.5 Identification and authentication (FIA)

5.1.5.1 Authentication Failure Management (NDcPP21:FIA_AFL.1)

NDcPP21:FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [**0-10**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password. (TD0408 applied)

NDcPP21:FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending remote Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***]. (TD0408 applied)

5.1.5.2 Password Management (NDcPP21:FIA_PMG_EXT.1)

NDcPP21: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.

5.1.5.3 Pre-Shared Key Composition (VPNGW10:FIA_PSK_EXT.1)

VPNGW10:FIA_PSK_EXT.1.1
The TSF shall be able to use pre-shared keys for IPsec and [***no other protocols***].

VPNGW10:FIA_PSK_EXT.1.2
The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and [***[between 6 and 160 characters]***];
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')').

VPNGW10:FIA_PSK_EXT.1.3
The TSF shall condition the text-based pre-shared keys by using [***[conversion from ASCII to binary for IPsec]***].

VPNGW10:FIA_PSK_EXT.1.4
The TSF shall be able to [***accept***] bit-based pre-shared keys.

5.1.5.4 Protected Authentication Feedback (NDcPP21:FIA_UAU.7)

NDcPP21: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.

5.1.5.5 Password-based Authentication Mechanism (NDcPP21:FIA_UAU_EXT.2)

NDcPP21:FIA_UAU_EXT.2.1
The TSF shall provide a local [***password-based, SSH public key-based [RADIUS and TACACS+ based username/password]***] authentication mechanism to perform local administrative user authentication. (TD0408 applied)

5.1.5.6 User Identification and Authentication (NDcPP21:FIA_UIA_EXT.1)

NDcPP21: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***].

NDcPP21: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.

5.1.5.7 X.509 Certificate Validation (NDcPP21:FIA_X509_EXT.1/Rev)

NDcPP21:FIA_X509_EXT.1.1/Rev
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 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, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***]
- 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.

NDcPP21: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.

5.1.5.8 X.509 Certificate Authentication (NDcPP21/VPNGW10:FIA_X509_EXT.2)

NDcPP21/VPNGW10: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***].

NDcPP21/VPNGW10:FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***allow the Administrator to choose whether to accept the certificate in these cases***].

VPNGW10 Application Note: FIA_X509_EXT.2.1: *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.*

5.1.5.9 X.509 Certificate Requests (NDcPP21/VPNGW10:FIA_X509_EXT.3)

NDcPP21/VPNGW10:FIA_X509_EXT.3.1

The TSF shall generate a Certification 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***].

NDcPP21/VPNGW10:FIA_X509_EXT.3.2

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

VPNGW10 Application Note: *The Base-PP defines this SFR as selection-based with its inclusion being dependent on the communications protocols that the TSF supports. Since a TOE that conforms to this PP-Module must support IPsec, this SFR is mandatory. Aside from mandating its inclusion in the TOE boundary, this PP-Module does not modify the SFR.*

5.1.6 Security management (FMT)

5.1.6.1 Management of security functions behavior (NDcPP21:FMT_MOF.1/Functions)

NDcPP21:FMT_MOF.1.1/Functions

The TSF shall restrict the ability to [***modify the behaviour of***] the functions [***transmission of audit data to an external IT entity***] to Security Administrators.

5.1.6.2 Management of security functions behaviour (NDcPP21:FMT_MOF.1/ManualUpdate)

NDcPP21:FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

5.1.6.3 Management of security functions behaviour (NDcPP21:FMT_MOF.1/Services)

NDcPP21:FMT_MOF.1.1/Services

The TSF shall restrict the ability to enable and disable start and stop services to Security Administrators.

5.1.6.4 Management of TSF Data (NDcPP21:FMT_MTD.1/CoreData)

NDcPP21:FMT_MTD.1.1/CoreData

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

5.1.6.5 Management of TSF data (NDcPP21/VPNGW10:FMT_MTD.1/CryptoKeys)

NDcPP21/VPNGW10: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].

VPNGW10 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.*

5.1.6.6 Specification of Management Functions (NDcPP21/ STFFW13/ VPNGW10:FMT_SMF.1)

NDcPP21/STFFW13/VPNGW10: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 and [***no other***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to configure firewall rules;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure the lifetime for IPsec SAs;
- Ability to import X.509v3 certificates to the TOE's trust store;
- Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this PP-Module; (**VPNGW10**)
- Ability to configure all security management functions of the TOE identified in other sections of this PP-Module (**VPNGW10**);
  [
  - ***Ability to start and stop services;***
  - ***Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;***
  - ***Ability to set the time which is used for time-stamps;***
  - ***Ability to configure NTP;***
  - ***Ability to configure the reference identifier for the peer***].

ST Application Note: *The FMT_SMF.1 requirement from the NDcPP21, STFFW13 and VPNGW10 have been combined to form a superset. The VPNGW10 modifies this SFR to make several of the functions mandatory instead of being selectable as they are in the NDcPP21. The STFFW13 adds the management function: ability to configure firewall rules.*

VPNGW10 Application Note: *The TOE is required to provide the ability to configure the packet filtering functionality that is specified by FPF_RUL_EXT.1 as well as the IPsec functionality that is specified by the Base-PP. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.*

5.1.6.7 Restrictions on Security Roles (NDcPP21:FMT_SMR.2)

NDcPP21:FMT_SMR.2.1
The TSF shall maintain the roles: - Security Administrator.
NDcPP21:FMT_SMR.2.2
The TSF shall be able to associate users with roles.
NDcPP21: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.

5.1.7 Packet Filtering (FPF)

5.1.7.1 Rules for Packet Filtering (VPNGW10:FPF_RUL_EXT.1)

VPNGW10:FPF_RUL_EXT.1.1
The TSF shall perform Packet Filtering on network packets processed by the TOE.

VPNGW10: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)
  o Source address
  o Destination Address
  o Protocol
- IPv6 (RFC 2460)
  o Source address
  o Destination Address
  o Next Header (Protocol)
- TCP (RFC 793)
  o Source Port
  o Destination Port
- UDP (RFC 768)
  o Source Port
  o Destination Port

VPNGW10:FPF_RUL_EXT.1.3
Refinement: The TSF shall allow the following operations to be associated with Packet Filtering rules: permit and drop with the capability to log the operation.

VPNGW10:FPF_RUL_EXT.1.4
The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

VPNGW10: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.

VPNGW10:FPF_RUL_EXT.1.6
The TSF shall drop traffic if a matching rule is not identified.

5.1.8 Protection of the TSF (FPT)

5.1.8.1 Protection of Administrator Passwords (NDcPP21:FPT_APW_EXT.1)

NDcPP21:FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form. (TD0483 applied)

NDcPP21:FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords. (TD0483 applied)

5.1.8.2 Fail Secure (Self-test Failures) - Self Test (VPNGW10:FPT_FLS.1/SelfTest)

VPNGW10: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].

5.1.8.3 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP21:FPT_SKP_EXT.1)

NDcPP21:FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.8.4 Reliable Time Stamps (NDcPP21:FPT_STM_EXT.1)

NDcPP21:FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

NDcPP21:FPT_STM_EXT.1.2
The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server***].

5.1.8.5 TSF testing (NDcPP21/VPNGW10:FPT_TST_EXT.1)

NDcPP21/VPNGW10:FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: noise source health tests, [

1. **ArubaOS OpenSSL Module:**
   a) **AES Known Answer Tests (KAT)**
   b) **Triple-DES KAT**
   c) **RNG KAT**
   d) **RSA KAT**
   e) **ECDSA (sign/verify)**
   f) **SHA (SHA1, SHA256 and SHA384) KAT**
   g) **HMAC (HMAC-SHA1, HMAC-SHA256 and HMAC-SHA384) KAT**
2. **ArubaOS Cryptographic Module**
   a) **AES KAT**
   b) **Triple-DES KAT**
   c) **SHA (SHA1, SHA256, SHA384 and SHA512) KAT**
   d) **HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT**
   e) **RSA (sign/verify)**
   f) **ECDSA (sign/verify)**
   g) **FIPS 186-2 RNG KAT**
3. **ArubaOS Uboot BootLoader Module**
   a) **Firmware Integrity Test: RSA 2048-bit Signature Validation**
4. **Aruba Hardware Known Answer Tests:**
   a) **AES KAT**
   b) **AES-CCM KAT**
   c) **AES-GCM KAT**
   d) **Triple DES KAT**
   e) **HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT**
5. **Conditional Self-tests:**
   a) **Continuous Random Number Generator Test**
   b) **Bypass Test**
   c) **RSA Pairwise Consistency Test**
   d) **ECDSA Pairwise Consistency Test**
   e) **Firmware Load Test**

   ].

VPNGW10 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.*

5.1.8.6 Extended: Self-Test with Defined Methods (VPNGW10:FPT_TST_EXT.3)

VPNGW10: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].

VPNGW10: FPT_TST_EXT.3.2
The TSF shall execute the self-testing through [a TSF-provided cryptographic service specified in FCS_COP.1/SigGen].

5.1.8.7 Trusted update (NDcPP21/VPNGW10:FPT_TUD_EXT.1)

NDcPP21/VPNGW10: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 [***no other TOE firmware/software version***].
NDcPP21/VPNGW10: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***].
NDcPP21/VPNGW10: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 mechanism***] prior to installing those updates.

VPNGW10 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.*

5.1.9 TOE access (FTA)

5.1.9.1 TSF-initiated Termination (NDcPP21:FTA_SSL.3)

NDcPP21:FTA_SSL.3.1
The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.9.2 User-initiated Termination (NDcPP21:FTA_SSL.4)

NDcPP21:FTA_SSL.4.1
The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.1.9.3 TSF-initiated Session Locking (NDcPP21:FTA_SSL_EXT.1)

NDcPP21:FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.9.4 Default TOE Access Banners (NDcPP21:FTA_TAB.1)

NDcPP21: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.

5.1.9.5 TOE Session Establishment (VPNGW10:FTA_TSE.1)

VPNGW10:FTA_TSE.1.1
The TSF shall be able to deny establishment of a remote VPN client session based on [location, time, day, [***[blacklist]]***].

ST Application Note: *Location is defined as the client's IP address. Blacklist refers to a list of denied clients identified by MAC address.*

5.1.9.6 VPN Clien

5.1.9.7 t Management (VPNGW10:FTA_VCM_EXT.1)

VPNGW10:FTA_VCM_EXT.1.1
The TSF shall assign a private IP address to a VPN client upon successful establishment of a security session.

ST Application Note: The private IP address is one that is internal to the trusted network for which the TOE is the headend.

5.1.10 Trusted path/channels (FTP)

5.1.10.1 Inter-TSF trusted channel (NDcPP21:FTP_ITC.1)

NDcPP21: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.

NDcPP21:FTP_ITC.1.2
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

NDcPP21:FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [

- **Transmitting audit records to an audit server**
- **Communicating with authentication servers**
- **Synchronizing time with an NTP server**

].

5.1.10.2 VPN Inter-TSF Trusted Channel (VPN Communications) - (VPNGW10:FTP_ITC.1/VPN)

VPNGW10: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.

VPNGW10:FTP_ITC.1.2/VPN
The TSF shall permit [the authorized IT entities] to initiate communication via the trusted channel.

VPNGW10:FTP_ITC.1.3/VPN
The TSF shall initiate communication via the trusted channel for [***remote VPN gateways/peers***].

VPNGW10 Application Note: *The FTP_ITC.1 requirement in the Base-PP relates to other trusted channel functions. This iteration is specific to IPsec VPN communications.*

5.1.10.3 Trusted Path (NDcPP21:FTP_TRP.1/Admin)

NDcPP21:FTP_TRP.1.1/Admin
The TSF shall be capable of using [***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.

NDcPP21:FTP_TRP.1.2/Admin
The TSF shall permit remote Administrators to initiate communication via the trusted path.

NDcPP21:FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic Functional Specification
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM Coverage
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 9 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d
The developer shall provide a functional specification.

ADV_FSP.1.2d
The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d
The developer shall provide operational user guidance.

AGD_OPE.1.1c
The operational user guidance shall describe, for each user role, the useraccessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c
The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c
The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c
The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c
The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6c
The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c
The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d
The developer shall provide the TOE, including its preparative procedures.

AGD_PRE.1.1c
The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c
The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e
The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d
The developer shall provide the TOE for testing.

AVA_VAN.1.1c
The TOE shall be suitable for testing.

AVA_VAN.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2e
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to

determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TOE generates audit records for start-up and shutdown of the TOE, all administrator actions, and for all the events identified in Table 8 Auditable Events which includes the entire packet contents for packets transmitted/received during IPsec peer session establishment. 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 column 3 of Table 8. For the administrative task of generating/import of, changing, or deleting of cryptographic keys, the key is uniquely identified in the audit records by the name assigned to it during import.

The TOE can be configured to log events by including the 'log' keyword when defining a firewall rule. In the event that a TOE network interface is overwhelmed by traffic the TOE will drop packets and generate an audit event for every packet that is denied and dropped.

The TOE stores audit records locally and provides CLI and Web UI capabilities for the administrator to view the contents of the audit trail. For local storage, the maximum log file size for all processes is ARUBA_MAX_LOG_FILE_SIZE (i.e 95,304 bytes). However, for 72xx and x86 platforms, the security.log, system.log and user-debug logs have a maximum file size given by A_MAX_SECURITY_USER_DEBUG_LOG_FILE_SIZE (i.e 349,524 bytes). The local protected log storage operates using the first in, first out (FIFO) method, therefore audit logs are overwritten when the available space is exhausted.

The TOE can also be configured to send audit records to a trusted third-party syslog server in the operational environment. The TOE uses IPsec to protect the communication channel between itself and the remote syslog server. If an external syslog server has been enabled, all audit logs are simultaneously written to both the local audit log and the syslog server. The local audit logs and logs sent to a remote server are identical.

The Security audit function is designed to satisfy the following security functional requirements:

- NDcPP21/STFFW13/VPNGW10:FAU_GEN.1: The TOE generates audit events for the not specified level of audit. Each audit record includes the date and time of the event, type of event, subject identity
- NDcPP21:FAU_GEN.2: 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.
- NDcPP21:FAU_STG.1: The TOE protects audit records in local storage from unauthorized modification or deletion. The local logs can only be viewed – they cannot be deleted or modified. There are no CLI or GUI commands for such actions.

- NDcPP21:FAU_STG_EXT.1: The TOE can be configured to export audit records to an external syslog server. The communication must be protected with an IPsec tunnel between the TOE and the syslog server. When the local storage space for audit data is full, the TOE will overwrite the previous audit records on a first in, first out (FIFO) basis.

6.2 Cryptographic support

The TOE includes two cryptographic modules that provide supporting cryptographic functions. The ArubaOS Crypto Module provides all of the cryptographic functions implemented for IKEv2/IPsec, while the ArubaOS OpenSSL Module provides all other cryptographic functions implemented in the TOE including those for IKEv1/IPsec, TLS and SSH.

The evaluated configuration requires that the TOE be configured in FIPS mode to ensure that the CAVP tested algorithms are used. The following functions have been CAVP certified:

Requirements	Functions	Standards	Cert	Cert
	Cryptographic key generation		Hardware Appliances (Broadcom XLP, Intel Atom (Denverton))	Virtual Appliances (Intel Xeon Gold, Intel Xeon D, Intel Atom, Intel Core i5, Intel Core i7, Intel Xeon E3)
FCS_CKM.1 FCS_CKM.1(1) [VPN]	RSA schemes using cryptographic key sizes of 2048-bit or greater	FIPS Pub 186-4	**C1972 C1974 C1229 C1230**	**A1015 A1020 C1732 C1731 C1972 C1974**
FCS_CKM.1 FCS_CKM.1(1) [VPN]	ECC schemes using 'NIST curves' P-256, P-384	FIPS Pub 186-4	**C1972 C1974 C1229 C1230**	**A1015 A1020 C1732 C1731 C1972 C1974**
FCS_CKM.1	FFC schemes using cryptographic key sizes of 2048-bit or greater	FIPS Pub 186-4	**C1972 C1974 C1229 C1230**	**A1015 A1020 C1732 C1731 C1972 C1974**
	Cryptographic key establishment/distribution			
FCS_CKM.2	RSA-based key establishment schemes	RFC 8017 PKCS #1	**Vendor Affirmed**	**Vendor Affirmed**
FCS_CKM.2	Elliptic curve-based key establishment schemes	NIST SP 800-56A	**C1972 C1974 C1229 C1230**	**A1015 A1020 C1732 C1731 C1972 C1974**

Requirements	Functions	Standards	Cert	Cert
FCS_CKM.2	Finite field-based key establishment schemes	NIST SP 800-56A	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
	Encryption/Decryption			
FCS_COP.1/ DataEncryption	AES CBC (128, 192 and 256 bits)	FIPS Pub 197 NIST SP 800-38A	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
FCS_COP.1/ DataEncryption	AES GCM (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38D	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
FCS_COP.1/ DataEncryption	AES CTR (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38D	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
	Cryptographic signature services			
FCS_COP.1/SigGen	RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS Pub 186-4	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
FCS_COP.1/SigGen	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve size of 256 or 384 bits	FIPS Pub 186-4	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
	Cryptographic hashing			
FCS_COP.1/Hash	SHA-1/256/384 (digest sizes 160, 256, and 384 bits)	FIPS Pub 180-3	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
	Keyed-hash message authentication			

Requirements	Functions	Standards	Cert	Cert
FCS_COP.1/ KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 (key and output MAC sizes 160, 256, and 384, respectively)	FIPS Pub 198-1 FIPS Pub 180-3	**C1972** **C1974** **C1229** **C1230**	**A1015** **A1020** **C1732** **C1731** **C1972** **C1974**
	Random bit generation			
FCS_RBG_EXT.1	CTR_DRBG (AES) with HW based noise sources (256 bits)	NIST SP 800-90	**C1974** **C1229**	**A1020** **C1731** **C1974**

Table 10 Cryptographic Functions

The TOE fulfills all of the FIPS PUB 186-4 requirements for cryptographic key generation without extensions. The TOE conforms to all shall, shall-not, should and should-not statements. For RSA key establishment, the TOE implements section B.3.6 in Appendix B.3 of FIPS PUB 186-4. For ECDSA, the TOE implements section B.4.2 in Appendix B of FIPS PUB 186-4. The TOE implementation of Diffie-Hellman group 14 (2048 MODP) meets RFC 3526, Section 3.

Security Function	Communication Type	Key Establishment Methods
Administration	TLS	RSA Schemes ECC Schemes FFC Schemes
Administration	SSH	DH-14
Trusted Channels for Syslog, NTP, Authentication Services	IPsec	RSA Schemes ECC Schemes DH-14

Table 11 Key Exchange Methods used by TOE Services

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 two hardware based noise sources: a hardware random number generator provided by a Trusted Platform Module chip (TPM RNG Entropy source) and the network processor that serves as the main CPU for the Mobility Controller (CPU RNG Entropy source). The third entropy source used for virtual controllers is the Jitter RNG Entropy source. The *jitterentropy* implementation is based upon noise gathered by measuring the jitter in CPU execution time. The timing for the execution of a fixed piece of code is unrepeatable in any machine. The Intel DRNG kernel replenishes the entropy pool by both *rngd* and *jitterentropy* daemons. There is a fourth entropy source that is software based and supplied by the Linux Kernel RNG, however, Aruba does not view this as a useful source of entropy as the only sources of software/kernel entropy events used in the Aruba modules are messages between the dataplane and control plane, and flash memory access. While they do feed the Linux Entropy pool, they do not significantly contribute to it. Aruba's use of the Linux kernel RNG is therefore primarily for building up and storing entropy from hardware noise sources.

The TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. Zeroization is accomplished by overwriting the secret or private key with all zeroes. The table below 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.

CSP	CSPs type	Generation and Use	Storage	Zeroization
Key Encryption Key (KEK) (*applies only to MCs. There is no KEK for CSPs in the VMCs)	Triple-DES (192 bits)	Hardcoded during manufacturing. Used only to protect keys stored in the flash, not for key transport.	Stored in Flash memory (plaintext).	Zeroized by using command 'zeorize-tpm-keys' for hardware or 'wipe out flash' for virtual
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 device
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 device
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 device
DRBG V	SP800-90a CTR_DRBG V (128 bits)	Internal V value used as part of SP800-90a CTR_DRBG	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the device
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 device
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 device
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 device

CSP	CSPs type	Generation and Use	Storage	Zeroization
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 device
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 device
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 device
RADIUS server shared secret	8-128 characters shared secret	Entered by CO role. Used for RADIUS server authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
Enable secret	8-32 characters password	Entered by CO role. Used for CO role authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
User Password	8-32characters password	Entered by CO role. Used for User role authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
IKEv1 Pre-shared secret	Shared secret (8 - 64 ASCII or 64 HEX characters)	Entered by CO role. Used for IKEv1 peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret

CSP	CSPs type	Generation and Use	Storage	Zeroization
skeyid	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was established via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving other keys in IKE protocol implementation.	Stored in SDRAM memory (plaintext)**.**	Zeroized by rebooting the device.
skeyid_d	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 (IKEv1). Used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
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 device
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 (IKEv1/IKEv2). Used for IKEv1/IKEv2 payload integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
IKE session encryption key	Triple-DES (192 bits, 3 Key, CBC) /AES (128/192/256 bits, CBC)	The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKE payload protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device

CSP	CSPs type	Generation and Use	Storage	Zeroization
IPSec session encryption key	Triple-DES (192 bits, 3 Key, CBC) / AES and AES-GCM (128/256 bits, CBC) NOTE: 192 bit CAVS tested, but not used.	The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used to secure IPsec traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
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 (IKEv1/IKEv2). Used to verify the integrity of IPsec traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device
SSHv2 session key	AES (128/192/256 bits)	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 device
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 device
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 device
TLS session encryption key	AES 128/192/256 bits	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 device
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 device

CSP	CSPs type	Generation and Use	Storage	Zeroization
RSA Private Key	RSA 2048 bit private key	This key is generated in the module. Used for IKEv1, IKEv2, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	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 CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv1, IKEv2, TLS, OCSP (verifying OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all'
ECDSA Private Key	ECDSA suite B P-256 and P-384 curves	This key is generated in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	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 entered by the CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all'.
Factory CA Public Key	RSA (2048 bits)	This is RSA public key. Loaded into the module during manufacturing. Used for Firmware verification.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all'
SNMPv3 authentication password	8-64 characters password	Entered by CO role. User for SNMPv3 authentication.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret

CSP	CSPs type	Generation and Use	Storage	Zeroization
			-encrypted with AES256 (VMCs)	
SNMPv3 engine ID	10 - 24 characters password	Entered by CO role. A unique string used to identify the SNMP engine.	Stored in Flash memory (ciphertext) -encrypted with AES256 using the KEK (MCs) -encrypted with AES256 (VMCs)	Zeroized by using command 'write erase all' or by overwriting with a new secret
SNMPv3 session key	AES-CFB key (128 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SNMPv3). Used to secure SNMPv3 traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the device

Table 12: CSPs

The supporting cryptographic functions are included to support the HTTPS/TLS (RFCs 2818, TLS 1.2 (RFC 5246), SSHv2 (RFCs 4251, 4252, 4253, 4254 and 4344), and IPsec (RFC 4301) secure communication protocol. In the FIPS mode of operation, the cipher parameters have been hardcoded to use only the Common Criteria evaluated configuration and are not configurable by the administrator. No optional protocol characteristics are implemented.

The TOE supports TLSv1.2. Any other SSL/TLS versions are not supported by the TOE and such connection attempts will be rejected. Remote administration via the Web UI is protected using TLS/HTTPS. The following cipher suites are implemented by the TOE, and other than identifying the cipher no configuration is needed to support any of these ciphers:

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

The TOE performs RSA key establishment with key sizes 2048 bits and generates DH parameters over NIST curve secp256r1.

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 and HMAC-SHA1-96 and RSA using the following key exchange methods: diffie-hellman-group14-sha1. The TOE also supports SSH_RSA for server authentication. While other ciphers and hashes are implemented in the product, they are disabled while the TOE is operating in FIPS mode.

SSHv2 supports both public-key and password-based authentication and can be configured. SSHv2 connections are rekeyed after a period of no longer than one hour or no more than one gigabyte of transmitted data. Both of these thresholds are checked by the TOE and rekeying is performed upon reaching the threshold that is hit first. Password based authentication can be configured. The authentication timeout period is 30 seconds allowing clients to retry only 3 times. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. 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.

The TOE includes an implementation of IPsec in accordance with RFC 4301 for security. The TOE supports IPsec in transport mode and tunnel mode. The IPsec ESP protocol is implemented in conjunction with AES-CBC-128, AES-CBC-192 and AES-CBC-256 (as specified by RFC 3602) and AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106) together with the HMAC-SHA-1 algorithm.

The TOE implements both IKEv1, as defined in RFCs 2407, 2408, 2409, and RFC 4109; and IKE2, with support for NAT traversal, as defined in RFC 5996 and RFC 4868 for hash functions (HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384). Diffie-Hellman (DH) Groups 14, 19, and 20 are supported for both IKEv1 and IKEv2 as are RSA and ECDSA certificates and pre-shared key IPsec authentication. The TOE uses the AES-CBC-128, AES-CBC-192 and AES-CBC-256 algorithms as specified in RFC 3602 to encrypt the IKEv1 or IKEv2 payload. Note that aggressive mode is not used with IKEv1, only main mode is supported.

The TOE generates the secret value x used in the IKEv1/IKEv2 Diffie-Hellman key exchange ('x' in gx mod p) using the FIPS validated RBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256 or 384 bits (for DH Groups 14, 19, and 20, respectively). The TOE supports the following PRF hash functions: PRF_HMAC_SHA1, PRF_HMAC_SHA256 and PRF_HMAC_SHA384 and generates nonces used in the IKEv1 and IKEv2 exchanges of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash. Nonces are generated using the function RANDOM_numberGenerator() which generates numbers that meet the requirements specified in FCS_RBG_EXT.1 for random bit generation.

The TOE supports SA lifetime limits based on length of time for both IKEv1 and IKEv2. Lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) and IKEv2 SAs are established during administrator configuration of the IKE policies. In the case of IKEv1, lifetimes can be configured based on length of time within 1-24 hours for phase 1 and within 1-8 hours for phase 2 by specifying the number of seconds for the SA lifetime. In the case of IKEv2, lifetimes for the IKEv2 IKE_SA can be configured by specifying the number of seconds within 1-24 hours. For the IKEv2 IKE_CHILD SA, lifetimes can be configured by specifying the number of seconds within 1-8hrs for the SA lifetime.

In the IKEv1 phase 1 and phase 2 and 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 the 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.

The Administrator is responsible for ensuring that IKE/IPsec policies are configured so that the strength of the negotiated symmetric algorithm (in terms of the number of bits in the key) in the IKEv1 Phase 2 SA or IKEv2 CHILD_SA is less than or equal to the strength of the IKEv1 Phase 1 SA or IKEv2 IKE_SA. Administrators should configure IKE/IPsec policies so that the strength of the IKE association is greater than or equal to the strength of the IPsec tunnel (for example, by always using AES-256). However, if a misconfiguration is made, the TOE will reject the security association along with generating an audit log message.

Pre-shared keys used for IPsec can be constructed of essentially any alphabetic character (upper and lower case), numerals, and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 6-160 characters in length (e.g., 22 characters). The TOE requires suitable keys to be entered by an authorized administrator using a Web GUI or CLI function.

The TOE will only establish a trusted IPsec channel if the presented identifier in the received certificate matches the configured reference identifier, where 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.

The TOE implements an SPD to determine what traffic gets protected with IPsec, what gets bypassed, and what gets dropped. The SPD is achieved via the routing table and firewall policies. For client-to-gateway VPN links, the firewall policies are in control of how traffic is forwarded. For site-to-site VPN, the routing table is in control. In each case, additional routing or firewall rules may be applied. The TOE administrator implicitly configures the IPsec SPD via the routing table and firewall policies and includes a final rule that causes the network packet to be discarded if no other rules are matched. Packet processing is described in section 6.4.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- NDcPP21:FCS_CKM.1, VPNGW10:FCS_CKM.1/IKE: The TOE supports cryptographic key generation for the following schemes: RSA schemes using key sizes of 2048 bits or greater, ECC schemes using NIST curves P-256 and P-384, FFC schemes using key sizes of 2048 bits or greater, FFC schemes using Diffie-Hellman group 14 cryptographic key derivation.
- NDcPP21:FCS_CKM.2: The TOE supports cryptographic key establishment for RSA based key establishment schemes, Elliptic curve based key establishment schemes, Finite field based key establishment schemes and key establishment schemes using Diffie-Hellman group 14.
- NDcPP21:FCS_CKM.4: Keys are zeroized when they are no longer needed by the TOE.
- NDcPP21:FCS_COP.1/DataEncryption: The TOE supports AES CBC (128, 192 and 256 bits) and AES GCM (128 and 256 bits) for data encryption/decryption.
- VPNGW10:FCS_COP.1/DataEncryption: The TOE supports AES CBC (128, 192 and 256 bits) and AES GCM (128 and 256 bits) for data encryption/decryption.
- NDcPP21:FCS_COP.1/SigGen: The TOE supports rDSA (modulus 2048) and ECDSA with elliptical curve size 256 or 384 bits for signature generation and verification.
- NDcPP21:FCS_COP.1/Hash: The TOE supports SHA-1/256/384 (digest sizes 160, 256, and 384 bits) for cryptographic hashing. The TOE implements SHA-1, SHA-256 and SHA-384 in support of TLS v1.2 in conjunction with AES (CBC and GCM) 128- or 256-bit ciphers and RSA and ECDSA.
- NDcPP21:FCS_COP.1/KeyedHash: The TOE supports HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 (key and output MAC sizes 160, 256, and 384, respectively) for keyed-hash message authentication. The TOE implements HMAC-SHA-1, HMAC-SHA-1-96 and diffie-hellman-group 14-sha1 in support of SSHv2 and HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 in support of IPsec.
- NDcPP21:FCS_HTTPS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions in compliance with RFC 2818. If an invalid peer certificate is presented, the TOE will not establish the connection.
- NDcPP21:FCS_IPSEC_EXT.1/VPNGW10:FCS_IPSEC_EXT.1: The TOE supports IPsec cryptographic network communication protection (RFC 4868, RFC 4945).
- NDcPP21:FCS_NTP_EXT.1: The TOE updates its system time using IPsec to provide a trusted communication path between itself and an NTP v4 server.
- NDcPP21:FCS_RBG_EXT.1: The TOE supports CTR_DRBG (AES) 256 bits with HW based noise sources for random bit generation.
- NDcPP21:FCS_SSHS_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- NDcPP21:FCS_TLSS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.

6.3 User Data Protection

Network packets are received in memory buffers pre-allocated at boot time. The buffers are populated in the network interface receive ring. When the CPU receives network packets from the network interface, the CPU allocates a free buffer from the preallocated pool and replenishes the receive ring. When the CPU has finished packet processing, the

CPU adds the memory buffer associated with this network packet to the free buffer pool. Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required.

Each pre-allocated memory buffer is fixed at 1792 bytes and is sufficient to hold a standard Ethernet frame. When a frame is received by the network interface, a proprietary header is prepended onto the frame indicating its length, among other parameters. The frame is then sent to the CPU. The CPU reads the length out of the proprietary header and uses this to process the frame out of the buffer. For standard Ethernet frames, only a single frame is placed into a memory buffer – buffers do not contain multiple frames.

Jumbo packets are supported – these must be split across multiple memory buffers. The proprietary header ensures that the packet is reconstructed correctly.

The length in the proprietary header is always correct – the product would not function if this were not the case.

The User data protection function is designed to satisfy the following security functional requirements:

- STFFW13:FDP_RIP.2: Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required (i.e., packet content is always overwritten before being read).

6.4 Firewall

The TOE performs stateful packet filtering. Filtering rules may be applied to appliance Ethernet interfaces and to user-roles (wireless clients connecting through APs are placed into user-roles). Stateful packet filter policies are applied to user-roles to allow fine grained control over wireless traffic.

The TOE is comprised of a data plane and a control plane. Network packets are processed in the data plane, which is the first component that is initialized. Network interfaces are not brought 'up' until initialization is complete and the data plane operating system is fully initialized. The control plane operating system (management interfaces) boots simultaneously.

All packet level enforcement is performed within the data plane. The data plane implements the stateful firewall policy and packet filtering configuration which is used to control information flow on the network. No traffic is passed until the data plane and all of the functions it implements are up and running. If a ruleset is applied to an interface, it will always be processed. In case of system error, component failure, or incoming traffic that exceeds the TOE's maximum threshold, packets are dropped by default effectively stopping traffic. The TOE can also be configured to detect an attack rate of packets per 30 seconds. If the TOE determines that it is under a DoS attack, it will silently start dropping packets.

The TOE supports stateful processing of the following protocols:

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

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

- ICMPv4
  - Source Address

- Destination Address
- Type
- Code[2]

- ICMPv6
  - Source Address
  - Destination Address
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
  - IPv6 Extension header type
- TCP
  - Source Address
  - Destination Address
  - Source Port
  - Destination Port
- UDP
  - Source Address
  - Destination Address
  - Source Port
  - Destination Port
  - Distinct interface

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

The TOE allows the stateful packet filtering rules to be assigned to each distinct network interface. The interfaces can be viewed through the CLI command "show interface". Stateful session tracking is established and maintained by the data plane operating system thorough inspection of network packets, including handshake transactions.

The algorithm applied to incoming packets is as follows:

[2] ICMPv4 Code 134 is an unsolicited router advertisement which is discarded/dropped by the TOE. All other traffic is handled based upon access control lists configured on the TOE.

- Check for IP fragments and assemble.
- Parse and identify protocol in the IP packet.
- Perform length checks and apply default rules.
- Enforce interface access-lists (ACLs) if configured.
- Lookup session. If exists, then apply corresponding session / stateful ACLs.
- Add session if it doesn't exist. (stateful if the protocol warrants, as listed above.). If stateful also open reverse ports for returning/stateful session. The protocol attributes identified above are used in session determination and will include IP protocol attributes for higher level protocols. TCP processing is further described below. Generate log message, if the 'log' keyword is configured in the rule.
- 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.

During TCP session establishment, the session is identified by source IP, destination IP, source port, and destination port. By default, the three-way handshake is not enforced before data is allowed. This behavior can be changed using the configuration "firewall enforce-tcp-handshake". Once the session has been established in the datapath session table, future packets which match the source IP, destination IP, source port, and destination port are processed by the "fast path" which was previously programmed during session establishment.

By default, TCP sequence numbers are ignored. This behavior can be changed using the configuration "firewall enforce-tcp-handshake" command which prevents data from passing between two clients until the three-way TCP handshake has been performed, at an upper limit of 8 half-open TCP connections. The "firewall enforce-tcp-sequence" command enforces the TCP sequence numbers for all packets. When applied, the TOE will monitor traffic sent through the TOE and drop half-open TCP connections.

TCP flags are largely ignored by the firewall, with the obvious exception of SYN/ACK during session establishment, and FIN/RST during session teardown.

Sessions are removed if the relevant protocol frame is received (e.g. TCP RST) or aged out after a configurable timeout of inactivity (whichever occurs first). Session removal is immediate. Audit log messages are not generated when a session is removed.

The TOE enforces the default stateful traffic filtering rules specified in FFW_RUL_EXT.1.6 and any administrator defined rules. These rules, called access-lists (ACLs), are in the sequence specified in the packet processing algorithm above. Only a single access-list may be applied to an Ethernet interface. Multiple access-lists may be applied to a user-role. If multiple access-lists are applied, they are processed in order from top to bottom. The first match found is selected and no further processing takes place. If no match is found, the default action is deny. If a rule is applied permitting or denying traffic, applying the inverse of this rule will overwrite the pre-existing rule. The TOE will not allow inverse rules to be applied.

The TOE is also capable of dropping and logging network packets according to the rules specified in FFW_RUL_EXT.1.7. The administrator must configure a default Access Control List in order to ensure that this traffic is dropped and logged.

The Firewall function is designed to satisfy the following security functional requirement(s):

- STFFW13:FFW_RUL_EXT.1: The TOE performs Stateful Traffic Filtering on network packets processed by the TOE.

6.5 Identification and authentication

The TOE supports role-based authentication. Users can authenticate to an external authentication server or to the Controller's internal database.

The administrator can create a user account in the internal database and assign a predefined role to that account. Users logging in to the Controller are restricted based on their assigned role. In this case, the authentication mechanism is provided by the TOE and the credentials are maintained in the internal database. The administrator can also configure the TOE so that users are authenticated using an external authentication server. The TOE supports RADIUS and TACACS+ servers. A trusted channel using IPsec is established between the TOE and an external authentication server.

Remote administrators are configured as users who have privileges to access the CLI and Web GUI administration interfaces and who are authenticated as users using the local database or an external authentication server. The remote administrators authenticate as users using a username and password via Web GUI and username/password or public key for SSH. The Web GUI interface provides a trusted path to connect to the TOE via HTTPS. The HTTPS interface uses a server RSA or ECDSA certificate which is stored on the TOE. SSH provides a trusted path to connect to the CLI. It uses SSH_RSA for public keys. Direct console to the CLI only supports username/password. A successful logon takes place when a recognized username/password combination is provided.

For users connecting with a VPN client, the IPsec/IKE VPN is established between the TOE and the client prior to the user authentication using pre-shared keys or certificates and can optionally authenticate to the external authentication server using a username and password. The external authentication server communicates success or failure of the authentication to the TOE.

The TOE accepts pre-shared keys for IPsec (IKEv1, IKEv2). It accepts bit-based pre-shared keys and accepts text-based PSKs that are transformed into bit-based PSKs. Text-based keys are conditioned by being directly converted to binary.

The controller maintains a counter of failed authentication attempts for a given administrative username within the past three minutes. An unsuccessful authentication attempt is detected when an invalid password or public key is entered for a valid username. If the failed authentication threshold is reached for a given username, that user account is locked out until the configured lock-out period has expired. The failed authentication threshold is enforced by the TOE and when using an external authentication server. In order to ensure that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, the serial port does not enforce account locking. Therefore, local administrators accessing the TOE via the local console will always have access.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- NDcPP21:FIA_AFL.1: After an administrator specified (0-10) number of failed attempts, the TOE will lockout (blacklist) the offending remote administrator and log the event. The offending administrator will remain locked out until the administrator configured lock-out period has expired. FIA_AFL.1 is enforced by the TOE and when using an external authentication server.
- NDcPP21:FIA_PMG_EXT.1: The TOE authentication mechanism provides configuration for minimum password length. The following calculation is based on the following facts:
  - Password is case-sensitive
  - A-Z, a-z, 0-9, !@#$%^&*()_+, and extended characters
  - Password minimum length is set to 8
  - Passwords have maximum lifetime and new passwords must contain a minimum of 4 character changes from the previous password

  Passwords must be at least eight characters long. Numeric, alphabetic (upper and lower case), and keyboard/extended characters can be used, which gives a total of 95 characters to choose from. An eight character password using all characters has 95^8 total possible combinations. The probability for a random attempt to succeed is therefore less than one in 1,000,000,000,000,000.

- VPNGW10:FIA_PSK_EXT.1: The TOE accepts between 6-160 character text based pre-shared keys (composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')')) for IPsec and IKEv1/IKEv2.
- NDcPP21:FIA_UAU.7: The TOE provides only obscured feedback to the administrative user while authentication is in progress at the local console by displaying an asterisk (*) for each character entered.
- NDcPP21:FIA_UAU_EXT.2: The TOE provides local accounts 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) administrators. The TOE shall invoke the correct authentication mechanism as configured by the administrator.
- NDcPP21:FIA_UIA_EXT.1: 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 allowing any other TSF-mediated actions on behalf of that user.
- NDcPP21:FIA_X509_EXT.1/Rev: The TOE protects, stores and allows authorized administrators to load X.509v3 certificates for use to support authentication for IPsec connections. Certificates are loaded into the controller using the "Certificate Manager" section of the Web-based user interface. The controller supports loading of certificates in PEM, DER, or PFX format. Private keys may be loaded onto the controller through a password-protected PFX file or may originate on the controller at the time a Certificate Signing Request (CSR) is generated.

  During runtime, certificates and private keys are stored in ramdisk, in volatile memory, in decrypted form. This allows private keys to be accessed rapidly for high network load conditions. When powered off, private keys are stored encrypted in non-volatile (flash) memory. The encryption method used is AES256 as described in the CSP table.

  Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE. The TOE validates certificates in accordance with RFC 5280 certificate validation rules. The TOE validates the revocation status of certificates using OCSP as specified in RFC 6960 and CRL as specified in RFC 5759, Section 5.

  Both CRL and OCSP checking are supported. The CRL support is not robust – the CRL itself must be manually loaded onto the controller; there is no auto-update ability. CRLs are limited to a maximum of 256 entries. OCSP responder servers must be configured in the revocation profile. The profile consists of the server URL, the responder cert (used to verify OCSP responses), and the action to take if the OCSP responder is non-responsive (permit or deny). If revocation checking is enabled, all certs in the chain except for the root are verified in order, starting with the peer cert and ending at the penultimate CA certificate. Certificate validation includes verification of the basicConstraints extension and the CA flag to determine whether they are present and set to TRUE. The check also ensure that the key usage extension is present. OCSP certificates must have the OCSP signing purpose in the extendedKeyUsagefield.
- NDcPP21:FIA_X509_EXT.2: The TOE uses X509v3 certificates to support authentication for IPsec. If the TOE cannot determine the validity of a certificate, the administrator has the option of choosing whether or not to accept the certificate.
- NDcPP21:FIA_X509_EXT.3/ VPNGW10:FIA_X509_EXT.3: 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.

6.6 Security management

The TOE provides the administrator role the capability to enable the management of TOE security attributes, TSF data and TOE security functions. The administrator can configure TOE security settings and policies using the Web

UI via HTTPS, or the command line interface via serial console locally or remotely using SSHv2. The Web GUI is a just a front-end to the CLI (i.e., calls the CLI internally). It provides a user-friendly interface for the administrator to manage the TOE. Every function that can be performed on the Web GUI, can also be performed using the CLI but not vice versa. However, every security management function claimed can be done either using the Web GUI or CLI.

The TOE supports role-based authentication. There are three types of roles: administrator[3] role, limited administrator role, and wireless user role. The limited administrator role and wireless user role are not TOE Security management roles. The administrator can manage the TOE using the Web GUI or CLI. Wireless clients cannot access the TOE through the Web GUI or CLI interfaces and, therefore, do not have access to the management functionalities of the TOE. The limited administrator role can perform non-security tasks only.

The administrator/remote administrator authenticates with a username and password via an HTTPS connection or via the interactive command line. Once the administrator is authenticated, the Mobility Controller provides management interfaces which can be used by the administrator to configure the TOE security functions. Local administrators can also use the CLI via a serial console (direct) connection to the TOE by using username and password. Remote administrators may use the Web GUI interface from the browser or may also use the CLI interface via an SSH protocol connection from an SSH client.

The TOE provides the administrator with capabilities to manage all security functions identified in this Security Target, including the following:

- Configure the access banner warnings
- Configure the session inactivity time before session termination
- Perform TOE updates and verify the updates using digital signature capability
- Configure the authentication failure settings
- Configure firewall and packet filtering rules (creation, modification and deletion)
- Configure the cryptographic functionality including modifying, deleting, generating and importing cryptographic keys and certificates for VPN operation
- Configure IPsec functionality including configuring the lifetime for IPsec SAs
- Configure TLS functionality
- Configure SSH functionality
- Configure and Import X509v3 Certificates
- Configure the reference identifier for the peer
- Set the time used for timestamps
- Configure (start and stop) TOE services
- Configure NTP
- Manage Users and roles – resetting passwords

The Security management function is designed to satisfy the following security functional requirements:

- NDcPP21:FMT_MOF.1/Functions: Only Security Administrators can modify the behavior of the transmission of audit data to an external audit server.
- NDcPP21:FMT_MOF.1/ManualUpdate: Only Security Administrators can enable the function to perform a manual update.

[3] Some Aruba documents may refer to as the “root” and/or “crypto officer” role. This role fulfills the role of ‘Security Administrator’ as identified in FMT_SMR.2

- NDcPP21:FMT_MOF.1/Services: Only Security Administrators can enable and disable TOE functions and services.
- NDcPP21:FMT_MTD.1/CryptoKeys/ VPNGW10:FMT_MTD.1/CryptoKeys: Only Security Administrators can manage the cryptographic keys and certificates used for VPN operations.
- NDcPP21:FMT_MTD.1/CoreData: Only Security Administrators can manage TSF data. There are no security functions available through any interfaces prior to administrator login. Non-administrative users do not have access to the TOE via the Web UI or the CLI, therefore, they do not have any access to the security functions of the TOE.
- NDcPP21:FMT_SMF.1/STFFW13:FMT_SMF.1/VPNGW10:FMT_SMF.1: The TOE provides administrative functions to manage all TOE security functions identified in this Security Target.
- NDcPP21:FMT_SMR.2: The TOE maintains the security role of Security Administrator who can manage the TOE both remotely and locally. The TOE supports role-based authentication. Administrators can make use of both local and remotely accessible administrator interfaces.

6.7 Packet Filtering

The TOE may be used 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. This functionality may be used with VPN clients or with other VPN gateways (i.e. site-to-site VPN).

As a VPN gateway, the TOE implements the IPsec protocol and the associated cryptographic and audit requirements which are described in Section 6.2. When used with VPN clients, the clients initiate the VPN connection. When used with other VPN gateways, the TOE or the other VPN gateway may initiate the VPN connection. The TOE assigns a private IP address (internal to the trusted network for which the TOE is the headend) to a VPN client upon successful establishment of a session.

The administrator may configure a time-out period after which inactive VPN client sessions will be terminated. The TOE may also be configured to deny establishment of VPN client sessions based on client location (IP address), time, day or blacklisted client MAC address.

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.

The Packet Filtering function is designed to satisfy the following security functional requirements:

- VPNGW10:FPF_RUL_EXT.1: The TOE shall perform Packet Filtering on network packets processed by the TOE.

Further details regarding the definition of a stateful packet filtering policy and the attributes that are used to define rules for the associated protocols can be found in Section 6.4.

6.8 Protection of the TSF

The TOE has an internal hardware clock that provides reliable time stamps used for auditing. Note that VMCs use Timestamp Counters (TSC) as the clock source through the hypervisor. The internal clock may be synchronized with a time signal obtained from an external NTP server. Note that the clock is used primarily to provide a timestamp for audit records, but is also used to support timing elements of cryptographic functions, certificate validity checks, session timeouts, and unlocking of administrator accounts locked as a result of authentication failure.

The TOE runs a suite of self-tests during power-up and periodically during operation, 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. An administrator can choose to reboot the TOE to perform power-up self-tests. The Mobility Controller runs the suite of FIPS 140-2 validated cryptographic module self-tests during start-up or on request from the administrator, including immediately after generation of a key (FIPS self-tests, including the continuous RNG test). The self-tests operate in the same manner on both the MCs and VMCs.

The following test are performed:

- 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 and SHA512) KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC- SHA512) KAT
  - RSA (sign/verify)
  - ECDSA (sign/verify)
  - FIPS 186-2 RNG KAT
- ArubaOS Uboot BootLoader Module
  - Firmware Integrity Test: RSA 2048-bit Signature Validation
- Aruba Hardware Known Answer Tests:
  - AES KAT
  - AES-CCM KAT
  - AES-GCM KAT
  - Triple DES KAT
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT

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.
- **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.
- **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.

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.

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.

The TOE uses IPsec, TLS, and SSH for all communications terminating at the TOE. Each of these protocols includes features to detect replayed data and the TOE will reject any such data that is received.

In order to update the firmware, the administrator can download the firmware file from the Aruba support website. Using TFTP, FTP, SCP, or HTTPS (Web UI only), the administrator can import the image into the controller.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- NDcPP21:FPT_APW_EXT.1: Passwords are not stored in plaintext. They are stored in flash using a SHA1 hash. Note that for ArubaOS, flash is an SCSI or IDE disk partition. On the VMCs, the disk is a file in the hypervisor.
- VPNGW10:FPT_FLS.1/SelfTest: In order to prevent entering an insecure state, the TOE will shutdown 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.
- NDcPP21:FPT_SKP_EXT.1: The TOE provides no interfaces that allow pre-shared, symmetric or private keys to be read. Section 6.2 describes how the pre-shared keys, symmetric keys and private keys are stored.
- NDcPP21:FPT_STM_EXT.1: The TOE relies on external time and date information either provided manually by the Security Administrator or through the use of an external trusted NTP server.
- NDcPP21/VPNGW10:FPT_TST_EXT.1, VPNGW10:FPT_TST_EXT.3: The TOE offers a suite of self-tests to verify the correct operation of the key generation and static TSF cryptographic data.. Firmware integrity tests verify the digital signature of the code image using RSA 2048 bit signature validation. Software images are cryptographically signed, and an image with an invalid signature will not be copied by the controller into the image partition. Similarly, a software image stored in the image partition through external means will be rejected by the hardware bootloader if the image signature is invalid.
- NDcPP21:FPT_TUD_EXT.1/VPNGW10:FPT_TUD_EXT.1: The TOE allows administrators to query the current version of its firmware/software and allows those administrators to manually initiate firmware/software updates. Prior to installing any update, the administrator can verify the digital signature of the update. Administrators can update the TOE executable code using image files manually downloaded from the Aruba support portal. The administrator may perform an update from either the WebUI or CLI.

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 support portal.

The update image is digitally signed using RSA 2048-bit signature validation. When an update is initiated, the TOE verifies the digital signature with the stored public root CA certificate which is programmed into the boot ROM or virtual boot ROM (ie. for the VMCs) of all Aruba products at the time of manufacturing. Upon successful verification, the TOE boots using the new image. Should verification fail, the TOE will enter into an error state. 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.

6.9 TOE access

Whether connecting to the CLI (locally or remotely) or web GUI, the TOE displays an advisory message when an administrator logs on. The message is configurable by TOE administrators.

The TOE terminates remote or local administrator sessions after session inactivity time exceeds a configurable session idle timeout. The session idle timeout is the maximum amount of time an administrator may remain idle.

All users may also terminate their own sessions at any time simply by logging off their session.

In order to limit access to the administrative functions, the TOE can be configured to deny remote VPN clients based on the time/date, IP address (location), as well as information retained in a blacklist. Firewall rules are used to restrict access and can be configured to blacklist clients when a rule is violated. Unlike the other properties, the blacklist is dynamically managed by the TOE identifying potentially undesirable network devices based on observed activities. If a device is actively identified in the blacklist, it cannot be used to connect to an administrative interface.

The TOE access function is designed to satisfy the following security functional requirements:

- NDcPP21:FTA_SSL.3: By default, the TOE will terminate remote interactive sessions after a configurable time interval of session inactivity.
- NDcPP21:FTA_SSL.4: Administrative users can log off at any time by issuing the applicable command.
- NDcPP21:FTA_SSL_EXT.1: Local inactive administrator sessions on the TOE are terminated, just like remote inactive administrator sessions, after the configured timeout period.
- NDcPP21:FTA_TAB.1: 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.
- VPNGW10:FTA_TSE.1[VPN]: The TOE can deny establishment of a remote VPN client session based on location, time, day, and blacklist state. Location is defined as the client's IP address. Blacklist refers to a list of denied clients identified by MAC address.
- VPNGW10:FTA_VCM_EXT.1: The TOE assigns a private IP address (internal to the trusted network for which the TOE is the headend) to a VPN client upon a successful establishment of a security session.

6.10 Trusted path/channels

The TOE provides trusted paths for remote administration and trusted channels for communication between itself and peers in the operating environment.

For remote administrators, the TOE uses HTTPS/TLS to offer secure remote web GUI-based administration and SSH to offer a secure remote administration CLI.

The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and the external authentication server, syslog server, NTP server, and other VPN Gateways (ie. site-to-site VPN). To configure the channels, the administrator uses the Security -> Advanced -> VPN panel of the Web GUI to create the host-to-host IPsec/IKE connections. All configuration settings must specify CAVP tested encryption algorithms as specified by the FCP_COP.1 requirements.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- NDcPP21:FTP_ITC.1: The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and an external authentication server, syslog server, and NTP server.
- VPNGW10:FTP_ITC.1/VPN: The TOE uses IPsec to provide a communication channel between itself and remote VPN gateway peers.
- NDcPP21:FTP_TRP.1/Admin: The TOE uses SSH, TLS/HTTPS to provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.