Aruba Remote Access Point Version 6.5.1-FIPS Security Target

Version 1.1
September 26, 2017

Prepared for:

Aruba, a Hewlett Packard Enterprise company

3333 Scott Blvd
Santa Clara, CA 95054

Prepared By:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

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 a software only application consisting of the Aruba Remote Access Point Version 6.5.1-FIPS. The TOE is being evaluated as an IPsec VPN client installed on an Aruba Remote Access Point.

The focus of this evaluation is on the TOE functionality supporting the claims in the Protection Profile for IPsec Virtual Private Network (VPN) Clients. The only capabilities covered by the evaluation are those specified in the aforementioned Protection Profile, all other capabilities are not covered in the evaluation. The security functionality specified in [VPNPP], includes protection of communications with the Aruba Master Controller, identification and authentication at the machine level when establishing the IPsec connection, ability to verify the source and integrity of updates to the TOE, specify client credentials and VPN gateways to use for connections, and specifies NIST-validated cryptographic mechanisms.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Aruba Remote Access Point Version 6.5.1-FIPS Security Target

ST Version – Version 1.1

ST Date – 9/26/2017

TOE Identification – Aruba Remote Access Point Version 6.5.1-FIPS

TOE Developer – Aruba, a Hewlett Packard Enterprise company

Evaluation Sponsor – Aruba, a Hewlett Packard Enterprise company

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- The ST is conformant to the *Protection Profile for IPsec Virtual Private Network (VPN) Clients,* version 1.4, 21 October 2013 [VPNPP]

The following NIAP Technical Decisions apply:

- TD0037: IPsec Requirement_DN Verification applies and is addressed in this ST.
- TD0053: Removal of FCS_IPSEC_EXT.1.12 Test 5 from VPN IPSEC Client v1.4 and is addressed in this ST.
- TD0079: RBG Cryptographic Transitions per NIST SP 800-131A Revision 1 and is addressed in this ST.

- TD0107: FCS_CKM - ANSI X9.31-1998, Section 4.1.for Cryptographic Key Generation and is addressed in this ST.TD0138: IPsec VPN Client Testing of SPD Rules and is addressed in this ST.
- TD0140: FCS_IPSEC_EXT.1.12, Test 1 - Importing of Private Key and Certificate

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - Part 3 Conformant

1.3 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 letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - 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 …").
  - Extended Requirements are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the ST needs. To ensure these requirements are explicitly identified, the ending "_EXT" is appended to the newly created short name and the component.
- The [VPNPP] uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Acronyms

AES	Advanced Encryption Standard
AIA	Asserted Identification Attributes
AP	Access Point
CC	Common Criteria
CLI	Command Line Interface
CSR	Certificate Signing Request

ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HSM	Hardware Security Module
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
OCSP	Online Certificate Status Protocol
PKEK	Private Key Encrypting Key
PP	Protection Profile
RAP	Remote Access Point
RNG	Random Number Generator
SA	Security Association
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Function
VPN	Virtual Private Network
WebUI	Web User Interface
WLAN	Wireless Local Area Network

2. TOE Description

The TOE is the Aruba Remote Access Point Version 6.5.1-FIPS. The Aruba Remote Access Point Version 6.5.1-FIPS is the operating system and application engine for the controller-managed Remote Access Points (RAP). The TOE is not a general-purpose VPN client that works with third-party VPN gateways. The TOE is a software only application that performs the functions of an IPsec VPN client installed on an Aruba Remote Access Point. The claimed functionality for this evaluation is limited to the security functionality for a VPN client as claimed in the [VPNPP].

An Aruba Master Controller is required to terminate the IPsec connections from the TOE. The underlying hardware and network on which the TOE resides is considered to be part of the environment.

The Aruba Remote Access Point Version 6.5.1-FIPS can be installed on the following platforms:

- RAP-108 Remote Access Point (RAP-108-USF1, HPE SKU JW269A)
- RAP-109 Remote Access Point (RAP-109-USF1, HPE SKU JW275A)
- AP-205H Access Point (AP-205H, HPE SKU JW167A)

This security target focuses on the IPsec VPN capabilities of the TOE. The TOE is installed on an Aruba Remote Access Point (RAP) and provides secure VPN access for remote locations to connect to an Aruba Master Controller. The VPN Client performs encryption and decryption of network packets in accordance with a VPN security policy negotiated between the VPN Client and the Aruba Master Controller. Remote users can use the same features as corporate office users. For example, voice over IP (VoIP) applications can be extended to remote sites while the servers and the PBX remain secure in the corporate office.

The Aruba Master Controller functions as the VPN Gateway and is not considered part of the evaluation.

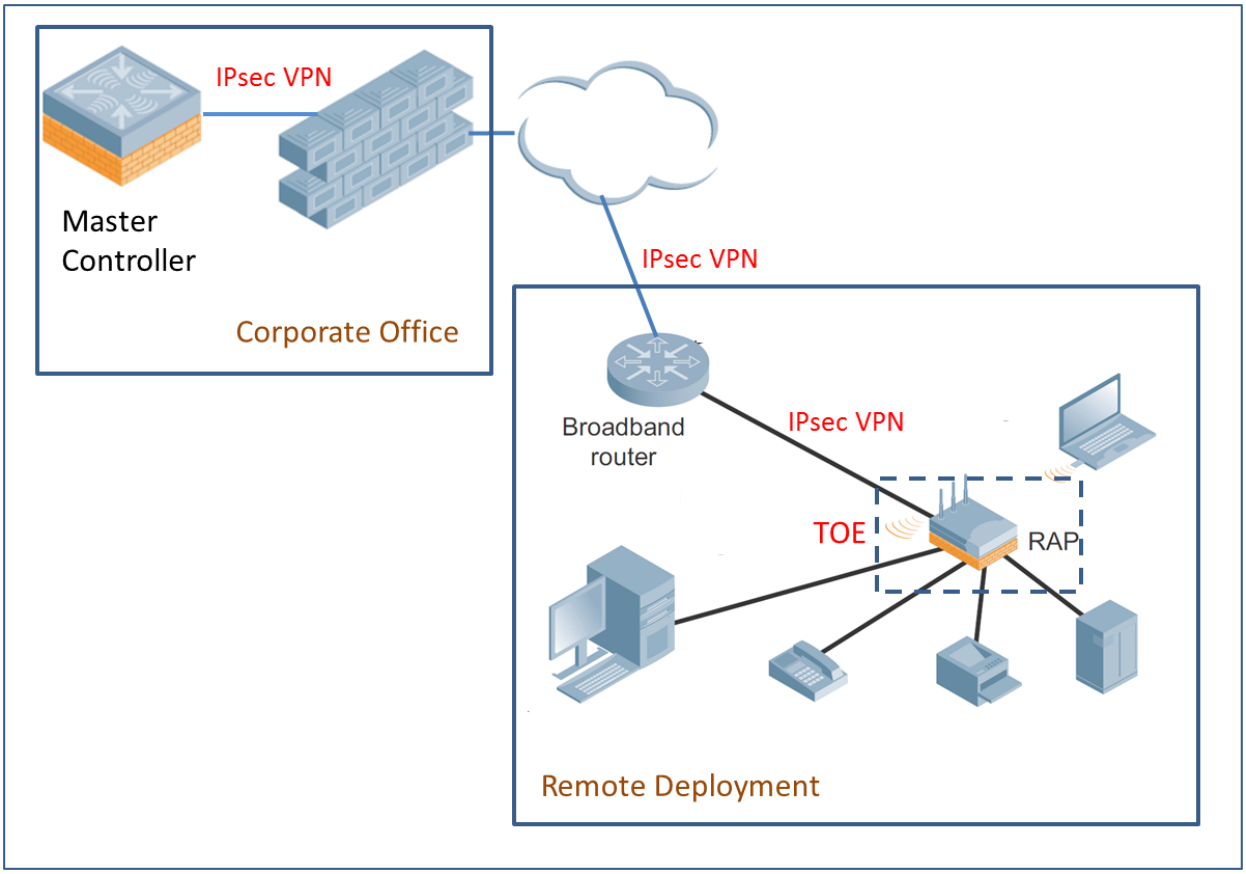

Figure 1 Example of Typical Remote VPN Client Deployment

2.1 TOE Overview

The TOE is the Aruba Remote Access Point Version 6.5.1-FIPS that is installed as an IPsec VPN client on an Aruba Remote Access Point.

In the CC evaluated configuration the TOE may be installed on the following platforms:

- RAP-108 Remote Access Point (RAP-108-USF1, HPE SKU JW269A)
- RAP-109 Remote Access Point (RAP-109-USF1, HPE SKU JW275A)
- AP-205H Access Point (AP-205H, HPE SKU JW167A)

The TOE must be configured to operate in the FIPS 140-2 Approved mode of operation. In FIPS-Approved mode, various weak protocols and algorithms are disabled.

2.2 TOE Architecture

The TOE is the Aruba Remote Access Point Version 6.5.1-FIPS which is the operating system and application engine for all Aruba controller-managed Remote Access Points (RAP). Designed for scalable performance, the TOE consists of three core components. First, a hardened, multicore, multithreaded supervisory kernel manages administration, authentication, logging and other system operation functions. This control plane is distinctly separate from the packet forwarding components to ensure continuous availability. Second, an embedded real-time operating system powers dedicated packet-processing hardware. This highly parallel architecture includes support for high-performance deep packet inspection of every connection that traverses the RAP, and implements all routing, switching and firewall functions. Third, a programmable encryption/decryption engine built on dedicated hardware delivers client-to-core encryption for wireless user data traffic and software VPN clients.

The claimed functionality for this evaluation is limited to the security functionality for a VPN client as claimed in the [VPNPP]. The security functionality specified in [VPNPP], includes protection of communications with the Aruba Master Controller, identification and authentication at the machine level when establishing the IPsec connection, ability to verify the source and integrity of updates to the TOE, specify client credentials and VPN gateways to use for connections, and specifies NIST-validated cryptographic mechanisms.

The TOE provided management functions are limited to specifying the IP address of the Aruba Master Controller, loading and managing certificates, and the identification of client credentials to be used for connections. Once the RAP is given the IP address of the Aruba Master Controller, it will bring up an IPsec tunnel using its factory-installed X.509 certificate (RSA2048/SHA1). When the RAP device boots it will establish a connection to its Aruba Master Controller and download its configuration file over a secure link. It can begin normal operation at this point, continuing to use the factory-installed certificate. All other management of the RAP is performed by the Aruba Master Controller.

Encryption and firewall policy can be configured and enforced from the IT department using profile based systems to insure uniformity. Troubleshooting can be performed on the packets entering the centralized location, and the RAP can be interrogated as to the state of connections and environmental conditions.

The Aruba Remote Access Point Version 6.5.1-FIPS allows users at any remote location equipped with a RAP to connect to an Aruba Master Controller over the Internet. These RAPs connect to the Aruba Master Controller using Layer-2 Tunneling Protocol and Internet Protocol Security (L2TP/IPsec) and send 802.11 data traffic through this tunnel. A secure RAP extends the corporate office to the remote site by giving remote users access to some of the same network features as corporate office users. They leverage the same access policies and service definitions used at headquarters or a branch office RAP deployment. Aruba Master Controllers act as VPN concentrators, eliminating the need for a parallel access infrastructure. For example, voice over IP (VoIP) applications can be extended to remote sites while the servers and the PBX remain secure in the corporate office.

The TOE implements NIST-validated cryptographic algorithms that support the IPsec protocols as well as digital signature services that support the secure update capabilities of the TOE. The encryption used to establish the secure IPsec VPN tunnel is provided by the TOE. Communication between the TOE and Aruba Master Controller uses the UDP 4500 port. IKE authentication can be configured to use digital certificates (RSA or ECDSA) to provide authentication.

2.2.1 Physical Boundaries

The TOE is the Aruba Remote Access Point Version 6.5.1-FIPS that is installed as an IPsec VPN client on an Aruba Remote Access Point. The Aruba Master Controller, the underlying hardware, and network on which the TOE resides are considered to be part of the environment.

The client is evaluated on the following platforms; each containing a TPM:

- RAP-108 Remote Access Point (RAP-108-USF1, HPE SKU JW269A)
- RAP-109 Remote Access Point (RAP-109-USF1, HPE SKU JW275A)
- AP-205H Access Point (AP-205H, HPE SKU JW167A)
-

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

The TOE protects itself from tampering and bypass through several mechanisms implemented by the TOE and the operating environment. The operating system is non-modifiable and the interfaces are strictly limited. An IPsec tunnel is used for all communications between the Aruba Master Controller and the RAP.

The sections below summarize the security functions provided by the TOE.

2.2.2.1 Cryptographic support

The TOE is a FIPS certified cryptographic module: the ArubaOS 6.5.1-FIPS (cert ##3021, #3023). The cryptographic module only employs FIPS-Approved DRBG, key generation, establishment, zeroization, encryption, digital signature, and hashing algorithms as specified by the FCS requirements.

2.2.2.2 User data protection

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

2.2.2.3 Identification and authentication

Remote authentication for the TOE is provided by RSA or ECDSA certificate-based RAP provisioning. The TOE supports Distinguished Name (DN) peer identifiers for certificate-based peer authentication. The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec sessions.

2.2.2.4 Security management

The administrator may configure the TOE via a WebUI or CLI interface on the RAP to specify the IP address of the Aruba Master Controller, loading and managing certificates, and the identification of client credentials to be used for connections in order to establish an IPsec VPN connection.

The TOE is managed by an administrator via the Aruba Master Controller (i.e. the VPN Gateway) to configure the VPN tunnel and all security functions identified in this Security Target.

2.2.2.5 Protection of the TSF

The TOE provides self-tests to ensure the correct operation of the cryptographic functions and TSF hardware. The TOE verifies the integrity of stored TSF executable code when it is loaded for execution.

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

2.2.2.6 Trusted path/channels

The TOE initiates an IPsec tunnel with the remote Aruba Master Controller.

2.3 TOE Documentation

Aruba Networks offers a series of documents that describe the installation and configuration of the Aruba Master Controller and Remote Access Points as well as guidance for subsequent use and administration of the applicable security features. The documentation is available online at http://support.arubanetworks.com. This section identifies the guidance documentation included in the TOE:

[USER]	ArubaOS 6.5.1.x User Guide, November 2016
[CLI]	ArubaOS 6.5.x Command-Line Interface Reference Guide, Revision 3, January 2017
[FIPS]	ArubaOS 6.5.1- FIPS Security Policy (available at CMVP website)
[CC_CONFIG]	Aruba VPN Client Protection Profile, Common Criteria Configuration Guide Version 1.4, June 2017
[Entropy]	Aruba Networks Aruba Remote Access Point Entropy Documentation, Revision 2.6, February 23, 2016

3. Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumption) from the [VPNPP].

In general, the [VPNPP] has presented a Security Problem Definition appropriate for remote users to use client appliances to establish an encrypted IPsec tunnel across an unprotected public network to a private network, and as such is applicable to the Aruba TOE.

4. Security Objectives

Like the Security Problem Definition, this security target includes by reference the Security Objectives from the [VPNPP]. The [VPNPP] security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

In general, the [VPNPP] has presented Security Objectives appropriate for a VPN to provide a protected transmission of private data between VPN Clients and VPN Gateways and as such are applicable to the Aruba Remote Access Point Version 6.5.1-FIPS TOE.

4.1 Security Objectives for the Environment

OE.NO_TOE_BYPASS

Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the operational environment.

OE.TRUSTED_CONFIG

Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

5. IT 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 Protection Profile (PP): *Protection Profile for IPsec Virtual Private Network (VPN) Clients*, version 1.4, 21 October 2013 [VPNPP]. The refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the [VPNPP] 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 the set of SARs specified in [VPNPP].

5.1 Extended Requirement Definitions

All of the extended requirements in this ST have been drawn from the [VPNPP]. The [VPNPP] defines the following extended SFRs and since they are not redefined in this ST, the [VPNPP] should be consulted for more information in regard to those CC extensions.

- FCS_CKM_EXT.2: Cryptographic Key Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (IPsec) Communications
- FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
- FIA_X509_EXT.1: Extended: X509 Certificate Validation
- FIA_X509_EXT.2: Extended: X509 Certificate Use and Management FPT_TST_EXT.1 Extended: TSF Self Test
- FPT_TUD_EXT.1 Extended: Trusted Update

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Aruba Remote Access Point Version 6.5.1-FIPS TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_CKM.1(1): Cryptographic Key Generation (Asymmetric Keys)
	FCS_CKM.1(2): Cryptographic Key Generation (for asymmetric keys - IKE)
	FCS_CKM_EXT.2: Cryptographic Key Storage
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (For Cryptographic Signature)
	FCS_COP.1(3): Cryptographic Operation (Cryptographic Hashing)
	FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)
	FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (IPsec) Communications
	FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
FDP: User data protection	FDP_RIP.2 Full Residual Information Protection

Requirement Class	Requirement Component
FIA: Identification and authentication	FIA_X509_EXT.1: Extended: X509 Certificate Validation
	FIA_X509_EXT.2: Extended: X509 Certificate Use and Management
FMT: Security management	FMT_SMF.1(1): Specification of Management Functions (Required of TOE)
	FMT_SMF.1(2): Specification of Management Functions (Provided by VPN Gateway)
FPT: Protection of the TSF	FPT_TST_EXT.1 Extended: TSF Self Test
	FPT_TUD_EXT.1 Extended: Trusted Update
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel

Table 1 TOE Security Functional Components

5.2.1 Cryptographic support (FCS)

5.2.1.1 Cryptographic Key Generation (Asymmetric Keys) (FCS_CKM.1(1))

FCS_CKM.1.1(1)

Refinement: The [***TOE***] shall generate asymmetric cryptographic keys used for key establishment in accordance with

- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;
- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [***no other curves***] (as defined in FIPS PUB 186-4, "Digital Signature Standard")
- [***no other***]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

5.2.1.2 Cryptographic Key Generation (for asymmetric keys - IKE) (FCS_CKM.1(2))

FCS_CKM.1.1(2)

Refinement: The [***TOE***] shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a:

[

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

5.2.1.3 Cryptographic Key Storage (FCS_CKM_EXT.2)

FCS_CKM_EXT.2.1

The [***TOE***] shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.2.1.4 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

Refinement: The [***TOE***] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

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

FCS_COP.1.1(1)

Refinement: The [***TOE***] shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm AES operating in GCM and CBC mode with cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, "Advanced Encryption Standard (AES)"
- NIST SP 800-38D, NIST SP 800-38A.

5.2.1.6 Cryptographic Operation (Cryptographic Signature) (FCS_COP.1(2))

FCS_COP.1.1(2)

Refinement: The [***TOE***] shall perform cryptographic signature services in accordance with a specified cryptographic algorithm:

- [***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA scheme***
- ***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 curve***]]

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

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

FCS_COP.1.1(3)

Refinement: The [***TOE***] shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384***] and message digest sizes [***160, 256, 384***] bits that meet the following: FIPS Pub 180-4, "Secure Hash Standard."

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

FCS_COP.1.1(4)

Refinement: The [***TOE***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384***], key size [**160, 256, 384**], and message digest size of [***160, 256, 384***] bits that meet the following: FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-4, "Secure Hash Standard".

5.2.1.9 Extended: Internet Protocol Security (IPsec) Communications (FCS_IPSEC_EXT.1)

FCS_IPSEC_EXT.1.1

The [***TOE***] shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2

The [***TOE***] shall implement [***tunnel mode***].

FCS_IPSEC_EXT.1.3

The [***TOE***] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4

The [***TOE***] shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC 4106, [***AES-CBC-256 (~~both~~ specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC***].

FCS_IPSEC_EXT.1.5

The [***TOE***] shall implement the protocol: **[*IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307*]**, and [***RFC 4868 for hash functions***].

FCS_IPSEC_EXT.1.6

The [***TOE***] shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and **[*no other algorithm*]**

FCS_IPSEC_EXT.1.7

The [***TOE***] shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Application Note: *The TOE implements IKEv2 and does not support IKEv1. This is permitted in [VPNPP].*

FCS_IPSEC_EXT.1.8

The [***TOE***] shall ensure that [***IKEv2 SA lifetimes can be configured by* [*VPN Gateway*]** based on **[*length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs***].

FCS_IPSEC_EXT.1.9

The [***TOE***] shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in gx mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**224 (for DH Group 14), 256 (for DH Group 19), 384 (for DH Group 20)**] bits.

FCS_IPSEC_EXT.1.10

The [***TOE***] shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^**[112 (for DH Group 14), 2^128 (for DH Group 19), and 2^192 (for DH Group 20)]**.

FCS_IPSEC_EXT.1.11

The [***TOE***] shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [***20 (384-bit Random ECP)*, *no other DH Groups***].

FCS_IPSEC_EXT.1.12

The [***TOE***] shall ensure that all IKE protocols perform peer authentication using a [***RSA, ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***no other method***].

FCS_IPSEC_EXT.1.13

The TOE shall support peer identifiers of the following types: [***Distinguished Name (DN)*]** and [***no other reference identifier type***].

FCS_IPSEC_EXT.1.14

The TOE shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

FCS_IPSEC_EXT.1.15

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

5.2.1.10 Extended: Cryptographic Operation: Random Bit Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The [***TOE***] shall perform all deterministic random bit generation services in accordance with [***NIST Special Publication 800-90A using [CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [***a platform-based RBG***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.2.2 User data protection (FDP)

5.2.2.1 Full Resident Information Protection (FDP_RIP.2)

FDP_RIP.2.1

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

5.2.3 Identification and authentication (FIA)

5.2.3.1 Extended: X509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

The [***TOE***] shall validate certificates in accordance with the following rules:

- Perform RFC 5280 certificate validation and certificate path validation.
- Validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560].***
- Validate the certificate path by ensuring the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.
- Validate the extendedKeyUsage field according to the following rules:
    - Certificates used for [***no other purpose***] shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3)

FIA_X509_EXT.1.2

The [***TOE***] shall only treat a certificate as a CA certificate if the following is met: the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.3.2 Extended: X.509 Certificate Use and Management (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

When a connection to determine the validity of a certificate cannot be established, the [***TOE***] shall [***not accept the certificate***].

FIA_X509_EXT.2.3

The [***TOE***] shall not establish an SA if a certificate or certificate path is deemed invalid.

5.2.4 Security management (FMT)

5.2.4.1 Specification of management functions (FMT_SMF.1(1)) (Required of TOE)

FMT_SMF.1.1(1)

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

- Specify VPN gateways to use for connections,
- Specify client credentials to be used for connections,
- [**Loading/managing certificates**].

5.2.4.2 Specification of management functions (FMT_SMF.1(2)) (Provided by VPN Gateway)

FMT_SMF.1.1(2)

The [***VPN Gateway***] shall be capable of performing the following management functions:

- Configuration of IKE protocol version(s) used,
- Configure IKE authentication techniques used,

- Configure the cryptoperiod for the established session keys. The unit of measure for configuring the cryptoperiod shall be no greater than an hour[1],
- Configure certificate revocation check,
- Specify the algorithm suites that may be proposed and accepted during the IPsec exchanges,
- load X.509v3 certificates used by the security functions in this PP,
- ability to update the TOE, and to verify the updates,
- configuration of the peer reference identifiers[2],
- ability to configure all security management functions identified in other sections of this PP,
- [***no other actions***].

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Extended: TSF Self Test (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

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

FPT_TST_EXT.1.2

The [***TOE***] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [**cryptographic service specified in FCS_COP.1(3)].**

5.2.5.2 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The [***TOE***] shall provide the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2

The [***TOE***] shall provide the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

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

5.2.6 Trusted path/channels (FTP)

5.2.6.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The [***TOE***] shall use IPsec to provide a trusted communication channel between itself and a VPN Gateway that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The [***TOE***] shall permit the TSF to initiate communication via the trusted channel.

FTP_ITC.1.3

The [***TOE***] shall initiate communication via the trusted channel for all traffic traversing that connection.

---

[1] The TOE has hard-coded values for Phase 1 (8 hours) and Phase 2 (2 hours) SA lifetimes for the cryptoperiod for the established session keys. NIAP has determined that this implementation is acceptable.

[2] FMT_SMF.1(2) has been modified to comply with TD 0037.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the [VPNPP].

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 2 Assurance Components

6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

The Aruba Remote Access Point Version 6.5.1-FIPS TOE provides all of the cryptographic functionality and meets FIPS 140-2 requirements by allowing the administrator to enable a FIPS operating mode. The FIPS 140-2 certified Aruba Remote Access Point Version 6.5.1-FIPS (CMVP ##3021, #3023) includes the following cryptographic implementations: ArubaOS UBoot Module; the ArubaOS OpenSSL Module; and the ArubaOS Crypto Module. The ArubaOS OpenSSL and ArubaOS Crypto Module are used for IPsec session cryptography, while the ArubaOS UBoot Module provides the cryptographic module integrity test on boot of the device; a requirement of the power-on self-tests.

The CC evaluated configuration of the TOE requires the use of this FIPS operating mode. In this mode, only FIPS-approved algorithms are allowed for cryptographic services (e.g., encryption, hashing, digital signature, etc.). All use of cryptographic services (IPsec/IKE) can only utilize FIPS-approved algorithms for the underlying algorithms.

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

Functions	Standards	Certificates
Asymmetric key generation		
• For key establishment purposes	NIST Special Publication 800-56A	KAS # 111, 116
• For IKE peer authentication	FIPS 186-4	RSA #1376, 1379, 2417, 1614, 2419 ECDSA # 466, 469, 581 DSA #1167, 1190
Encryption/Decryption		
• AES CBC and GCM (128-256 bits)	FIPS PUB 197 NIST SP 800-38A NIST SP 800-38D	Cert # 2450, 2677, 2680, 2689 [RAP-108 and RAP-109], #3176, 3177 [AP-205H]
Cryptographic signature services		
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-4	Cert # 1376, 1379,2417 [RAP-108 and RAP-109], #1614, 2419 [AP-205H]
• Elliptic Curve Digital Signature Algorithm (ECDSA) (P-256 and P-384)	FIPS PUB 186-4	Cert # 466, 469 [RAP-108 and RAP-109], #581 [AP-205H]
Cryptographic hashing		
• SHA-1, SHA-256, and SHA-384 (digest sizes 160, 256, and 384 bits)	FIPS Pub 180-4	Cert # 2246, 2249, 3654 [RAP-108 and RAP-109] , #2630, 3657 [AP-205-H]
Keyed-hash message authentication		
• HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 (digest sizes	FIPS Pub 198-1 FIPS Pub 180-4	Cert # 1663, 1666 [RAP-108 and RAP-109],

160, 256, and 384 bits)		#2005 [AP-205H]
Random bit generation		
• DRBG with a platform-based noise source of 256 bits	NIST Special Publication 800-90 using [CTR_DRBG(AES)	Cert # 433 [RAP-108 and RAP-109], #660 [AP-205H]

Table 3 Cryptographic Functions

Name	CSPs type	Generation	Storage and Zeroization	Use
Key Encryption Key	Triple-DES 168-bit key	Hardcoded during manufacturing	Stored in Flash. Zeroized by using command 'ap wipe out flash'	Encrypts IKEv2 ECDSA private key and configuration parameters.
DRBG entropy input	SP800-90a DRBG (512 bits)	Derived using NDRNG	Stored in plaintext in volatile memory. Zeroized on reboot.	DRBG initialization
DRBG seed	SP800-90a DRBG (384 bits)	Generated per SP800-90A using a derivation function	Stored in plaintext in volatile memory. Zeroized on reboot.	DRBG initialization
DRBG Key	SP800-90a (256 bits)	Generated per SP800-90A	Stored in plaintext in volatile memory. Zeroized on reboot.	DRBG
DRBG V	SP800-90a (128 bits)	Generated per SP800-90A	Stored in plaintext in volatile memory. Zeroized on reboot.	DRBG
Diffie-Hellman private key	Diffie-Hellman private key (224 bits)	Generated internally during Diffie-Hellman Exchange	Stored in the volatile memory. Zeroized after the session is closed.	Used in establishing the session key for an IPsec session

Name	CSPs type	Generation	Storage and Zeroization	Use
Diffie-Hellman public key	Diffie-Hellman public key (2048 bits)	Generated internally during Diffie-Hellman Exchange	Stored in the volatile memory. Zeroized after the session is closed.	Used in establishing the session key for an IPsec session
Diffie-Hellman shared secret	Diffie-Hellman shared secret (2048 bits)	Established during Diffie-Hellman Exchange	Stored in plain text in volatile memory, Zeroized when session is closed.	Used in establishing the session key for an IPsec session
EC Diffie-Hellman private key	Elliptic Curve Diffie-Hellman (P-256 and P-384).	Generated internally during EC Diffie-Hellman Exchange	Stored in the volatile memory. Zeroized after the session is closed.	Used in establishing the session key for an IPsec session
EC Diffie-Hellman public key	Elliptic Curve Diffie-Hellman (P-256 and P-384).	Generated internally during EC Diffie-Hellman Exchange	Stored in the volatile memory. Zeroized after the session is closed.	Used in establishing the session key for an IPsec session
EC Diffie-Hellman shared secret	Elliptic Curve Diffie-Hellman (P-256 and P-384)	Established during EC Diffie-Hellman Exchange	Stored in plaintext in volatile memory. Zeroized when session is closed.	Key agreement in IKEv2
IKEv2 session authentication key	HMAC-SHA-1/256/384 (160 / 256 / 384 bits)	Established as a result of IKEv2 service implementation.	Stored in plaintext in volatile memory. Zeroized when session is closed.	IKEv2 payload integrity verification
IKEv2 session encryption key	AES (128/256 bits)	Established as a result of IKEv2 service implementation.	Stored in plaintext in volatile memory. Zeroized when session is closed.	IKEv2 payload encryption

Name	CSPs type	Generation	Storage and Zeroization	Use
IPsec session encryption keys	AES (128/256 bits)	Established during the IPsec service implementation	Stored in plaintext in volatile memory. Zeroized when the session is closed.	Secure IPsec traffic
IPsec session authentication keys	HMAC-SHA-1 (160 bits)	Established during the IPsec service implementation	Stored in plaintext in volatile memory. Zeroized when the session is closed.	IPsec traffic authentication
RSA Private Key	RSA 2048 bits private key	Generated in the module.	Stored in non-volatile memory (Trusted Platform Module). Zeroized by physical destruction of the module.	Used by IKEv2 for device authentication
RSA public key	RSA 2048 bits public key	Generated in the module.	Stored in non-volatile memory. Zeroized by physical destruction of the module.	Used by IKEv2 for device authentication
ECDSA Private Key	ECDSA suite B P-256 and P-384 curves	Generated in the module	Stored in flash memory encrypted with KEK. Zeroized by the CO command **ap wipe out flash**.	Used by IKEv2 for device authentication.
ECDSA Public Key	ECDSA suite B P-256 and P-384 curves	Generated in the module	Stored in flash memory encrypted with KEK. Zeroized by the CO command ap wipe out flash.	Used by IKEv2 for device authentication.
Factory CA Public Key	RSA 2048 bits public key	Generated outside the module.	Stored in non-volatile memory. Zeroized by physical destruction of the module.	Firmware verification

Table 4 Critical Security Parameters

6.1.1 FCS_CKM.1(1): Cryptographic Key Generation (Asymmetric Keys)

The TOE implements a random number generator for finite field-based key establishment schemes, and for elliptic curve-based key establishment (conformant to NIST SP 800-56A). While the TOE generally fulfills all of the NIST SP 800-56A requirements without extensions, the following table specifically identifies the "should", "should not", and "shall not" conditions from those publications along with an indication of how the TOE conforms to those conditions.

NIST SP800-56A Section Reference	Requirement/ Recommendation Qualifier	Implemented?	Rationale for deviation
5.4	should	yes	Not applicable
5.5.1.1	should	yes	Not applicable
5.5.2	should	yes	Not applicable
5.6.2	should	yes	Not applicable
5.6.2.1	should	yes	Not applicable
5.6.2.2	should	yes	Not applicable
5.6.2.3	should	yes	Not applicable
5.6.3.1	should	yes	Not applicable
5.6.3.2.1	should	yes	Not applicable
5.6.4.1	shall not	no	Not applicable
5.6.4.2	should	yes	Not applicable
5.6.4.2	shall not	no	Not applicable
5.6.4.3	should (first occurrence)	yes	Not applicable
5.6.4.3	should (second occurrence)	yes	Not applicable
5.8	shall not (first occurrence)	no	Not applicable
5.8	shall not (second occurrence)	no	Not applicable
6	should	yes	Not applicable
7	shall not (first occurrence)	no	Not applicable
7	shall not (second occurrence)	no	Not applicable
9	shall not	no	Not applicable

Table 5 NIST SP800-56A Conformance

6.1.2 FCS_CKM.1(2): Cryptographic Key Generation (For Asymmetric Keys – IKE)

The TOE has the ability to upload custom RSA and ECDSA certificates to a RAP to be used for IKEv2 negotiation which establishes a tunnel between the RAP and the Aruba Master Controller. The TOE generates asymmetric keys for IKE peer authentication in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", and Appendix B.3 for RSA schemes. The TOE also implements the P-256 and P-384 ECDSA curves in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. The cryptographic key sizes are equivalent to, or greater than, a symmetric key strength of 112 bits.

6.1.3 FCS_CKM_EXT.2: Cryptographic Key Storage

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. The Diffie-Hellman and EC Diffie-Hellman private keys are zeroized after the session is closed. When powered off, IKEv2 ECDSA private keys are stored encrypted in non-volatile (flash) memory of the TOE platform. The RSA Private Key is stored encrypted in non-volatile memory (Trusted Platform Module). Persistent keys stored in both flash and the TPM are encrypted using the private key from TPM and the AES128 algorithm. The PKEK (TPM private key) is ultimately protected by hardware (TPM) on the underlying platform.

6.1.4 FCS_CKM_EXT.4: Cryptographic Key Zeroization

The TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. This function has also been subject to FIPS 140 certification. Zeroization is accomplished by executing the command "write erase all" which over writes zeros on all non-volatile memory in the product (flash and file system). Volatile keys are erased on power cycle, or they are overwritten when they are no longer needed (for ephemeral keys). The default RSA and the Factory CA Public Keys are stored in the TPM on the RAP. The TPM should be physically destroyed to wipe out the keys. **Table 4** identifies the key zeroization for the critical security parameters.

6.1.5 FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption)

The cryptographic algorithms used by the TOE include AES-CBC-128 and AES-CBC-256 (as specified by RFC 3602) and AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106) along with IKE2 as defined in RFCs 5996 and 4307.

6.1.6 FCS_COP.1(2): Cryptographic Operation (For Cryptographic Signature)

The TOE performs cryptographic signature services in accordance with the following specified cryptographic algorithms: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA scheme and FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384. RSA 2048 with HMAC-SHA-1 and ECDSA signing and verification is used to authenticate to the module during IKEv2. Both P-256 and P-384 curves are supported in establishing the session key for an IPsec session. RSA2048 and SHA-256 are used to sign the trusted update images. The cryptographic key sizes are equivalent to or greater than a symmetric key strength of 112 bits.

6.1.7 FCS_COP.1(3): Cryptographic Operation (Cryptographic Hashing)

The TOE performs cryptographic hashing services in accordance with the cryptographic algorithms SHA-1, SHA-256, SHA-384 that meet the FIPS Pub 180-4 "Secure Hash Standard". The SHA hash algorithm is used as part of HMAC, but is also used independently as part of digital signature creation and verification. The TOE generates RSA and ECDSA signatures during IKE peer authentication. The TOE verifies RSA & ECDSA signatures during IKE peer authentication. HMAC-SHA1/256/384 is used for the IKEv2 payload integrity verification.

6.1.8 FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)

The TOE performs keyed-hash message authentication with the HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 cryptographic algorithms. The HMAC-SHA-1 has a key size of 160 bits and a message digest size of 160 bits. Though the size of the output of HMAC-SHA-1 is the same as that of the underlying hash function (160 bits in the case of SHA-1), it can be truncated if desired to 96 bits (HMAC-SHA1-96) which complies with RFC 2104. The HMAC-SHA-256 has a key size of 256 bits and a message digest of 256 bits with a block size of 512 bits. The HMAC-SHA-384 has a key size of 384 bits and a message digest of 384 bits with a block size of 1024bits.

6.1.9 FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (IPsec) Communications

The TOE includes an implementation of IPsec in accordance with RFC 4301 for security. The protected communication between the TOE and Aruba Master Controller is provided by an IPsec VPN in tunnel mode.

The TOE implements various HMAC algorithms to be used for authentication with ESP and IKE. The specific algorithms used depend upon the ciphersuite being used. The TOE uses AES-GCM-128, AES-GCM-256 as specified in RFC 4106 and AES-CBC-256 specified by RFC 3602 as ESP encryption algorithms and implements HMAC-SHA1 as the authentication algorithm. The TOE uses AES-CBC-128 and AES-CBC-256 as the IKEv2 payload encryption algorithms and implements HMAC-SHA1, HMAC-SHA-256 and HMAC-SHA-384 as the authentication algorithm.

RSA 2048 with HMAC-SHA-1 and ECDSA suite B P-256 and P-384 curves are used by IKEv2 for device authentication.

The TOE supports IPsec cryptographic network communication protection (RFC 4868, RFC 4945). The gateway can be configured with various SA lifetimes, but it does not push the configuration down to the client. Rather, the TOE employs hard-coded values for Phase 1 (8 hours) and Phase 2 (2 hours) SA lifetimes. The gateway enforces its

SA lifetime configuration when the values configured on the gateway are less than the hard-coded values on the client. When the client values are less than the values on the gateway, the client will initiate rekeying for Phase 2 after 2 hours and for Phase 1 after 8 hours. In any case, the SAs are rekeyed within the limits specified in FCS_IPSEC_EXT.1.8.

The TOE uses its FIPS validated RBG specified in FCS_RBG_EXT.1 to generate the secret value "x" for each of its DH groups, having possible lengths of 224 (for DH Group 14), 256 (for DH Group 19), and 384 (for DH Group 20) bits. The TOE generates nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[112, 128, and 192]. The random number generated for the length of "x" and the nonces meet the stipulations in the [VPNPP].The IKEv2 protocols implemented by the TOE include DH Groups 14 (2048-bit MODP), 19 (256-bit ECP), and 20 (384-bit ECP) and utilize RSA (aka rDSA) or ECDSA peer authentication.

The TOE compares the peer's presented identifier (in the peer's X.509 certificate) to its reference identifier and fails the connection if they do not match.

In the IKEv2 IKE_SA and IKE_CHILD exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE initiates IKE negotiation, the DH group is sent in order according to the peer's configuration. When the TOE receives an IKE proposal, it will select the first match and the negotiation will fail if there is no match. The TOE default behavior insures that the strength of the negotiated symmetric algorithm negotiated for IKEv2 IKE_SA is greater than or equal to key strength negotiated for IKEv2 CHILD_SA. The only algorithm negotiated will be AES, therefore the strength is based solely upon the key size where more bits is the stronger strength.

The TOE provides mechanisms to implement an IPsec Security Policy Database (SPD) and to process packets to satisfy the behavior of DISCARD, BYPASS, and PROTECT packet processing as described in RFC 4301. This is achieved through the use of a split tunnel in conjunction with administrator configured access control lists (ACL). The TOE compares packets against the rules in the ACL to determine if any of the packets match the rules. The packets can be matched based upon source IP address, destination IP address, protocol type (e.g. TCP, UDP, ICMP). Traffic not matching any rule is passed to the next stage of processing. The TOE includes a final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

6.1.10 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

Entropy source (FCS_RBG_EXT.1.2) is described in a separate document, "Aruba Mobility Controller Entropy Documentation" prepared according to [VPNPP] Annex E. See **Table 3** Cryptographic Functions and **Table 4** Critical Security Parameters above.

The TOE makes use of the Linux kernel PRNG as a store of high-quality entropy gathered from the TOE platform TPM hardware chip. A TOE software process reads the output from the TPM and dumps it into the Linux kernel through /dev/random. All crypto libraries then use the /dev/random as their seed for CTR_DRBG.

Based on the TOE platform hardware design, the noise source provides 0.985785 bits of minimum entropy per binary digit output. In the normal operation, each approved RNG used in the module is seeded with 64 bytes (512 bits) of entropy inputs from the Linux entropy pool, yielding 512 * 0.985785 = 504.72 bits of entropy per seeding operation, which is larger than 256 bits entropy claimed in the assessment.

6.2 User data protection

All traffic in the communication path between the Aruba Master Controller and the RAP uses the IPsec tunnel. The TOE is designed to ensure that no residual information exists in network packets.

6.2.1 FDP_RIP.2 Full Residual Information Protection

The TOE is designed to ensure that no residual information exists in network packets. Previous packet content is made unavailable. The buffers are overwritten upon allocation with the new packet data. The 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.

6.3 Identification and authentication

The RAP needs to be provisioned before it is deployed at the remote location by the configuration of the IP address of the Aruba Master Controller. The RAP will establish an IPsec tunnel using its factory-installed X.509 certificate (RSA2048/SHA1). Once the communication to the Aruba Master Controller is established, the RAP will download a configuration file. There is no username/password involved in this process, nor does a person provisioning the RAP need to configure IPsec settings. The default IPsec profile is used for initial provisioning and subsequent settings are downloaded from the Aruba Master Controller. The administrator may also load a custom RSA or ECDSA certificate.

Remote authentication for the TOE is provided by RSA or ECDSA certificate-based RAP provisioning. For certificate-based peer authentication, the TOE supports Distinguished Name (DN) peer identifiers. The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec sessions.

6.3.1 FIA_X509_EXT.1: Extended: X509 Certificate Validation

The TOE uses X.509v3 certificates as defined in RFC 5280 to support authentication for IPsec connections. OCSP is used for obtaining the revocation status of an X.509 digital certificate. The RAP will pull the AIA field out of the certificate to determine how to reach an OCSP responder. The OCSP check occurs before the IPsec tunnel is established, therefore the OCSP responder must be "publically" available (e.g. on the Internet). Checking is also done for the basicConstraints extension and the cA flag to determine whether they are present and set to TRUE. If they are not, the certificate is not accepted.

6.3.2 FIA_X509_EXT.2: Extended: X509 Certificate Use and Management

Once the TOE is given the IP address of the Aruba Master Controller, the RAP will bring up an IPsec tunnel using its factory-installed X.509 certificate (RSA2048/SHA1). Once the Aruba Master Controller is reached in this way, the RAP will download a configuration profile. Note there is no username/password involved here, nor does a person provisioning the RAP need to configure IPsec settings. A default IPsec profile is used for initial provisioning, and subsequent settings are downloaded from the Aruba Master Controller. The RAP can begin normal operation at this point, continuing to use the factory-installed certificate. The TOE also provides an option to locally load a custom RSA or ECDSA certificate onto the RAP. As Suite-B mandates using the AES-GCM encryption and ECDSA certificates for security, this feature permits the upload of custom RSA and ECDSA certificates to a RAP. This allows custom certificates to be used for IKEv2 negotiation which establishes a tunnel between the RAP and the Aruba Master Controller. Once an administrator-generated certificate (ECDSA or RSA) is loaded, this credential will always be used for authentication. A RAP may contain only a single administrator-generated certification. If the administrator-generated certificate is later deleted, the RAP will again use the factory-installed certificate. The factory-installed certificate may not be deleted.

Feature support includes the ability to:

- Upload a single CA certificate and RAP certificate which have either elliptical crypto key parameters with ECDSA or RSA parameters for signing and verification.
- Store the certificate in the flash of the RAP
- Delete certificates
- Generate a CSR paired with a private key generation for the RAP. The private key is stored in the flash and the CSR can be exported out of the RAP to get it signed by the CA.

Certificates can only be imported using the RAP WebUI by navigating to the Certificates tab and uploading the certificate.

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec exchanges. The TOE validates the full path of X.509v3 certificates by validating the entire certificate chain from the leaf certificate up to the trusted root CA. The trusted root certificate and any intermediate certificates are known as the trust chain. These certificates must be loaded into the RAP flash.
The TOE checks the validity of the certificates in the path (the certificates stored in flash). Assuming the certificates are valid, the TOE finally checks the revocation status of all certificates. The TOE will reject any certificate for

which it cannot determine the validity and reject the connection attempt. The TOE will accept the certificate and allow secure SA if the response is "Good", it will reject the certificate and disallow secure SA if the response is revoked. The connection will always be denied if the OCSP server cannot be reached.

6.4 Security management

6.4.1 FMT_SMF.1(1): Specification of Management Functions (TOE)

The following security management functions are provided by the TOE. The TOE provides the administrator role the capability to enable the management of security attributes, TSF data and security functions, and specify client credentials (i.e., X.509v3 certificate) to be used for IKE connections. A WebUI and CLI interface exists on the RAP for local management for loading/managing certificates and basic troubleshooting. The administrator configures the TOE to specify the IP address of the VPN Gateway in order to establish an IPsec VPN connection.

6.4.2 FMT_SMF.1(2): Specification of Management Functions (VPN Gateway)

The Aruba Master Controller (i.e. the VPN Gateway) provides the following security management functions:

- Configuration of IKE protocol version used[3]
- Configure certificate based IKE authentication
- Configure certificate revocation check[4]
- Specify the algorithm suites that may be proposed and accepted during the IPsec exchanges,
- Load X.509v3 certificates used for VPN connections using IPsec
- Ability to update the TOE, and to verify the updates. The Aruba Master Controller is configured to automatically permit updates of the TOE software. Upon connecting to the Aruba Master Controller, the TOE checks for a software update. If an update is available, the TOE initiates the software download. The administrator can load valid TOE software updates onto the Controller.
- Configuration of the peer reference identifiers.
- Ability to configure all security management functions identified in other sections of the [VPNPP].

6.5 Protection of the TSF

The TOE runs a suite of self-tests during power-up which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. The TOE uses IPsec to provide a trusted communication channel between itself and a VPN Gateway in order to receive software updates. The software images are hashed and cryptographically signed.

6.5.1 FPT_TST_EXT.1: Extended: TSF Self Test Extended

The TOE executes a suite of self-tests during power up to verify the correct operation of the key generation and static TSF cryptographic data. In the event any self-test fails, the module enters an error state, logs the error, and reboots automatically. Software images stored in the image partition are hashed, and a software image will be rejected by the bootloader if the image hash is invalid. SHA-256 is used to verify the integrity of the image. On bootup, the RAP performs a SHA-256 hash of the Aruba Remote Access Point Version 6.5.1-FIPS image file and compares it to the hash included with the driver. Upon successful verification that the hash matches the one computed with the code the TOE will continue to load. If the integrity of stored TSF executable code fails, the TOE enters an error state, logs the error, and reboots automatically.

The TOE's FIPS certified ArubaOS 6.5.1 cryptographic module (cert #, #3021, #3023) performs the following power-up self-tests:

- ArubaOS OpenSSL Module Known Answer Tests:

[3] TOE only supports IKEv2 and no configuration is required
[4] No configuration is required to enable OCSP – it is enabled by default.

  - AES (encrypt/decrypt) KATs
  - Triple-DES (encrypt/decrypt) KATs
  - DRBG KAT
  - RSA KAT
  - ECDSA Sign/Verify
  - SHS (SHA1, SHA256, SHA384 and SHA512) KATs
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KATs
- ArubaOS Crypto Module Known Answer Tests:
  - AES (encrypt/decrypt) KATs
  - Triple-DES (encrypt/decrypt) KATs
  - SHS (SHA1, SHA256, SHA384 and SHA512) KATs
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KATs
  - RSA KAT
  - ECDSA Sign/Verify
- ArubaOS Uboot Bootloader Module Known Answer Tests:
  - Firmware Integrity Test: RSA PKCS#1 v1.5 (2048 bits) signature verification with SHA-1
- ArubaOS AP Kernel Crypto Module Known Answer Tests:
  - AES (encrypt/decrypt) KATs
  - AES-GCM KAT

These tests are sufficient to verify that the correct version of the TOE software is running as well as that the cryptographic operations are all performing as expected.

6.5.2 FPT_TUD_EXT.1: Extended: Trusted Update

The Aruba Master Controller is configured to automatically permit updates of the TOE software. Upon connecting to the Aruba Master Controller, the TOE checks for a software update. If an update is available, the TOE initiates the software download. When the download completes, the TOE sends a message to the Aruba Master Controller, informing it that the TOE has either successfully downloaded the new software version, or that the preload has failed for some reason. If the download fails, the TOE will retry the download after a brief waiting period. A software image that is downloaded from the Aruba Master Controller is both verified at the time of receipt (before writing to the flash) and is also verified by the bootloader each time the TOE boots. The software images are hashed and cryptographically signed, and an image with an invalid signature will not be copied by the RAP into the image partition. RSA2048 and SHA256 are used to sign the image. The signing certificate is issued by Aruba's internal CA. This CA is stored offline with private keys protected by an HSM. The public root CA is stored in the TOE's boot flash at the time of manufacturing. Upon verification, the software image is written to the flash and the TOE automatically reboots. The Local Debugging tab in the WebUI permits the user to display the device information, including the software version currently running on the TOE.

6.6 Trusted path/channels

The TOE uses the IPsec/IKE protocol with certificates to establish a trusted channel between itself and the Aruba Master Controller.

6.6.1 FTP_ITC.1: Inter-TSF Trusted Channel

The TOE uses IPsec to provide a trusted communication channel between itself and a VPN Gateway 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. Aruba Remote Access Point Version 6.5.1-FIPS supports the Layer 2 Tunneling Protocol (L2TP) with IPsec VPN to create a VPN tunnel from an Aruba Remote Access Point (RAP) to an Aruba Master Controller at the data center or headquarters.

7. Protection Profile Claims

The ST conforms to the *Protection Profile for IPsec Virtual Private Network (VPN) Clients*, version 1.4, 21 October 2013 [VPNPP].

As explained previously, the security problem definition, security objectives, and security requirements have been drawn verbatim from the [VPNPP].

8. Rationale

This security target includes by reference the [VPNPP] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [VPNPP] assumptions. [VPNPP] security functional requirements have been reproduced with the protection profile operations completed. Operations on the security requirements follow [VPNPP] application notes and assurance activities. Consequently, [VPNPP] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 6 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	Trusted path/channels
FCS_CKM.1(1)	X					
FCS_CKM.1(2)	X					
FCS_CKM_EXT.2	X					
FCS_CKM_EXT.4	X					
FCS_COP.1(1)	X					
FCS_COP.1(2)	X					
FCS_COP.1(3)	X					
FCS_COP.1(4)	X					
FCS_IPSEC_EXT.1	X					
FCS_RBG_EXT.1	X					
FDP_RIP.2		X				
FIA_X509_EXT.1			X			
FIA_X509_EXT.2			X			
FMT_SMF.1(1)				X		
FMT_SMF.1(2)				X		
FPT_TST_EXT.1					X	
FPT_TUD_EXT.1					X	
FTP_ITC.1						X

Table 6 Security Functions vs. Requirements Mapping