Hewlett Packard Enterprise

Aruba OpenSSL Module

Firmware Version 1.0

Non-Proprietary Security Policy

FIPS 140-3 Level 1

Document Version 1.0

November 2024

Copyright

© 2024 Hewlett Packard Enterprise Company. Hewlett Packard Enterprise Company trademarks include airwave , HPE Networks®, HPE Wireless Networks®, HPE Networking, the registered HPE Networking the Mobile Edge Company logo, HPE Networking Mobility Management System®, Mobile Edge Architecture®, People Move. Networks Must Follow®, RFProtect®, Green Island®. All rights reserved. All other trademarks are the property of their respective owners. HPE Networking is a Hewlett Packard Enterprise company.

The resource assets in this firmware may include abbreviated and/or legacy terminology for HPE Networking products. See https://www.hpe.com/us/en/networking/ for current and complete HPE Networking product lines and names.

Open Source Code

Certain Hewlett Packard Enterprise Company products include Open Source software code developed by third parties, including software code subject to the GNU General Public License (GPL), GNU Lesser General Public License (LGPL), or other Open Source Licenses. The Open Source code used can be found at this site:

https://myenterpriselicense.hpe.com/cwp-ui/software

Legal Notice

The use of Hewlett Packard Enterprise Company switching platforms and software or firmware, by all individuals or corporations, to terminate other vendors' VPN client devices constitutes complete acceptance of liability by that individual or corporation for this action and indemnifies, in full, Hewlett Packard Enterprise Company, from any and all legal actions that might be taken against it with respect to infringement of copyright on behalf of those vendors.

Hewlett Packard
Enterprise

https://www.hpe.com/us/en/networking/

1701 E Mossy Oaks Rd,
Spring, TX, USA 77389
Phone: 1-888-342-2156

Contents

Figures

Tables

Preface

Document Revision History

The following table lists the history of the revisions of this document by version number and date of revision.

Table 1 – Document Revision History

Version	Date	Description
1.0	November 2024	Initial FIPS 140-3 release for the Hewlett Packard Enterprise *Aruba OpenSSL Module* firmware version 1.0 used by ArubaOS firmware versions running on Hewlett Packard Enterprise hardware and virtual appliances

1 General

This section describes:

- The purpose of this document.
- Hewlett Packard Enterprise documents related to this document contents.
- Where to go for additional Hewlett Packard Enterprise product information.
- Acronyms and abbreviations.
- The assurance security levels for each of the areas described in the FIPS 140-3 Standard.

1.1 Purpose of this Document

This release supplement provides information regarding the Hewlett Packard Enterprise *Aruba OpenSSL Module* firmware version 1.0 FIPS 140-3 Level 1 validation from Hewlett Packard Enterprise (HPE). Throughout this document, references to HPE Networking are to the Hewlett Packard Enterprise division. The material in this supplement modifies the general Hewlett Packard Enterprise firmware documentation included with this product and should be kept with your Hewlett Packard Enterprise product documentation.

This supplement primarily covers the non-proprietary Cryptographic Module Security Policy for the Hewlett Packard Enterprise *Aruba OpenSSL Module* firmware version 1.0. This security policy describes how the module meets the security requirements of FIPS 140-3 Level 1 and how to place and maintain the module in the secure Approved mode. This policy was prepared as part of the FIPS 140-3 Level 1 validation of the product.

FIPS 140-3 (Federal Information Processing Standards Publication 140-3, *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. FIPS 140-3 aligns with ISO/IEC 19790:2012(E) and includes modifications of the Annexes that are allowed to the Cryptographic Module Validation Program (CMVP), as a validation authority. The testing for these requirements will be in accordance with ISO/IEC 24759:2017(E), with the modifications, additions or deletions of vendor evidence and testing allowed as a validation authority under paragraph 5.2. More information about the FIPS 140-3 standard and validation program is available on the National Institute of Standards and Technology (NIST) website at:

https://csrc.nist.gov/projects/cryptographic-module-validation-program

In addition, in this document, the Hewlett Packard Enterprise *Aruba OpenSSL Module* is referred to as the module, the cryptographic module, and Aruba OpenSSL module.

1.2 Additional Hewlett Packard Enterprise Product Information

More information is available from the following sources:

- See the Hewlett Packard Enterprise web site for the full line of products from HPE Networking**:**

  https://www.hpe.com/us/en/networking/

- The NIST Validated Modules web site contains contact information for answers to technical or sales-related questions for the product:

  https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search

  Enter **Hewlett Packard Enterprise** in the Vendor field then select Search to see a list of FIPS validated Hewlett Packard Enterprise cryptographic modules.

  Select the Certificate Number for the Module Name ‘Aruba OpenSSL Module’.

1.3 Acronyms and Abbreviations

AES	**Advanced Encryption Standard**
AP	**Access Point**
CAVP	**Cryptographic Algorithm Validation Program**
CBC	**Cipher Block Chaining**
CCCS	**Canadian Centre for Cyber Security, a branch of CSE**
CLI	**Command Line Interface**
CMVP	**Cryptographic Module Validation Program**
CO	**Crypto Officer**
CPSec	**Control Plane Security protected**
CSE	**Communications Security Establishment**
CSP	**Critical Security Parameter**
DF	**Derivation Function**
EAP	**Extensible Authentication Protocol**
ECO	**External Crypto Officer**
EMC	**Electromagnetic Compatibility**
EMI	**Electromagnetic Interference**
ESV	**Entropy Source Validation**
FE	**Fast Ethernet**
GE	**Gigabit Ethernet**
GHz	**Gigahertz**
HMAC	**Hashed Message Authentication Code**
Hz	**Hertz**
IKE	**Internet Key Exchange**
IPsec	**Internet Protocol security**
KAT	**Known Answer Test**
KEK	**Key Encryption Key**
L2TP	**Layer-2 Tunnelling Protocol**
LAN	**Local Area Network**
LED	**Light Emitting Diode**
NTP	**Network Time Protocol**
OCSP	**Online Certificate Status Protocol**
PCT	**Pairwise Consistency Test**
PSP	**Public Security Parameter**
SFTP	**Secure File Transfer Protocol**
SHA	**Secure Hash Algorithm**
SNMP	**Simple Network Management Protocol**
SSP	**Sensitive Security Parameter**
SPOE	**Serial & Power Over Ethernet**
TEL	**Tamper-Evident Label**
TFTP	**Trivial File Transfer Protocol**
TPM	**Trusted Platform Module**
WLAN	**Wireless Local Area Network**

1.4 Security Levels

The Hewlett Packard Enterprise *Aruba OpenSSL Module* is intended to meet overall FIPS 140-3 Level 1 requirements as shown in the following table.

Table 2 – Security Levels

ISO/IEC 24759 Section 6 [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services, and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	1
8	Non-Invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-Tests	1
11	Life-Cycle Assurance	1
12	Mitigation of Other Attacks	N/A
Overall	**Overall Security Rating of the Module**	**1**

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Hewlett Packard Enterprise *Aruba OpenSSL Module* version 1.0 (also referred to as 'the module') is a firmware type cryptographic module that was validated under FIPS 140-3 Level 1 requirements and meets the claims made in this document. ArubaOS is the operating system for Hewlett Packard Enterprise Mobility Conductors, Mobility Controllers/Gateways, and controller-managed Hewlett Packard Enterprise Access Points (APs). The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware) is an Hewlett Packard Enterprise cryptographic module that provides cryptographic services for the ArubaOS operating system running on the Hewlett Packard Enterprise hardware-based equipment or Hewlett Packard Enterprise virtual appliances.

Module Type: Firmware

Module Embodiment: Multiple-chip Standalone

2.1.1 Cryptographic Module Boundary

The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware) is an Hewlett Packard Enterprise cryptographic module that provides cryptographic services for the ArubaOS operating system.

The cryptographic boundary for the Hewlett Packard Enterprise *Aruba OpenSSL Module* is defined as the module component within the Linux-based User Space. The physical perimeter is the production-grade enclosure of the hardware chassis of the Hewlett Packard Enterprise hardware device or Hewlett Packard Enterprise virtual appliance host.

The module is one of the components within the ArubaOS firmware package in electronic form and is installed automatically when a trusted and verified ArubaOS is booted on an Hewlett Packard Enterprise device. The Hewlett Packard Enterprise *Aruba OpenSSL Module* component includes the module shared libraries and the associated integrity check files (used for integrity tests):

- libcrypto.so.1.0.2s
- libcrypto.so.1.0.2s.hmac
- libssl.so.1.0.2s
- libssl.so.1.0.2s.hmac

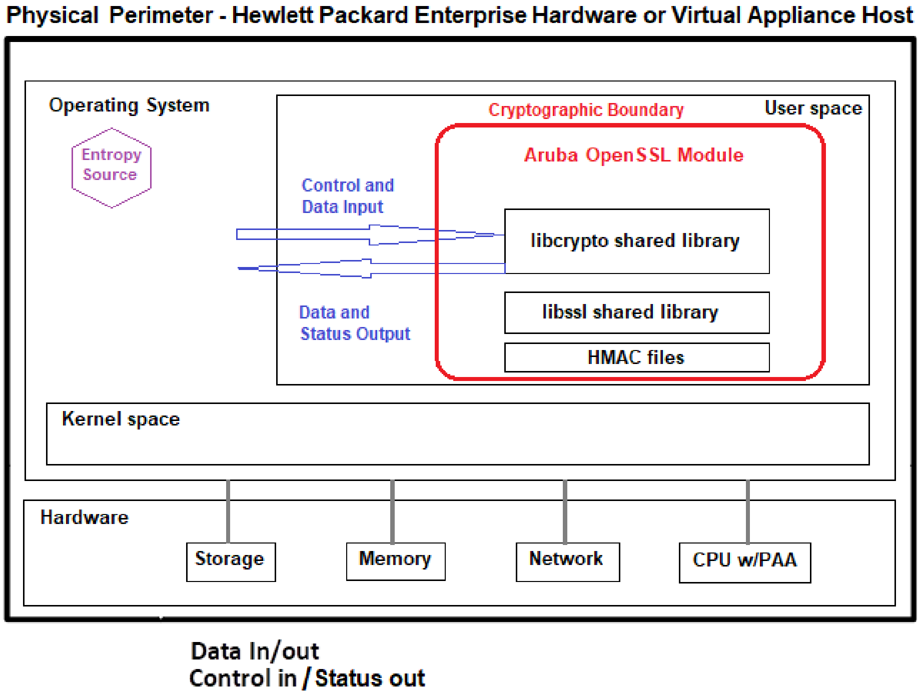

Figure 1 – Functional Block Diagram of Cryptographic Boundary for Aruba OpenSSL Module

2.2 Version Information

The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware) version 1.0 was validated against FIPS 140-3 Level 1 requirements. The CMVP makes no claim as to the correct operation of the module or the security strengths of the generated keys when operating under a version that is not listed on the validation certificate.

Table 3 – Version Information

Type	Versions	`'show ver'` Output
Firmware	Aruba OpenSSL Module version 1.0	`Aruba OpenSSL Module 1.0`

2.3 Operating Environments

The module operates in a limited operational environment. The module runs on the ArubaOS operating system and related hardware or virtual platform and provides cryptographic services for the ArubaOS operating system. See the following tables of Tested Operational Environments and Vendor Affirmed Operational Environments for details.

Table 4 – Tested Operational Environments

#	Operating System	Hardware / Virtual Platform	Processor	PAA / Acceleration
1	ArubaOS 8.10	7020 Mobility Controller	Broadcom XLP208 (MIPS64)	None
2	ArubaOS 8.10	7205 Mobility Controller	Broadcom XLP316 (MIPS64)	None
3	ArubaOS 8.10	7220 Mobility Controller	Broadcom XLP432 (MIPS64)	None
4	ArubaOS 8.10	7280 Mobility Controller	Broadcom XLP (MIPS64)	None
5	ArubaOS 8.10	9012 Gateway	Intel Atom C3508 (Denverton)	None
6	ArubaOS 8.10	9240 Gateway	Intel Xeon (Cascade Lake)	with / without PAA
7	ArubaOS 8.10	AP-505 Wireless Access Point	Broadcom BCM47622L (ARM-A7)	None
8	ArubaOS 8.10	AP-515 Wireless Access Point	Broadcom BCM (64-bit ARMv8)	None
9	ArubaOS 8.10	AP-535 Wireless Access Point	Qualcomm IPQ (64-bit ARM Cortex A53)	None
10	ArubaOS 8.10	AP-635 Wireless Access Point	Qualcomm IPQ (64-bit ARM Cortex A53)	None
11	ArubaOS 8.10	AP-655 Wireless Access Point	Qualcomm IPQ (64-bit ARM Cortex A53)	None
12	ArubaOS 8.10	MCR-HW-5K Mobility Conductor Hardware Appliance	Intel Xeon E5-2620v4 (Broadwell)	with PAA
13	ArubaOS 8.10 on VMWare ESXi 7.0	MCR-VA-50 Mobility Conductor Virtual Appliance on HPE Edgeline 20	Intel Xeon Gold 6212U (Cascade Lake)	None
14	ArubaOS 8.10 on VMWare ESXi 7.0	MC-VA-50 Mobility Controller Virtual Appliance on HPE ProLiant ML110 Gen10	Intel Xeon E3 1515 (Skylake)	None
15	ArubaOS 8.10 on VMWare ESXi 7.0	MC-VA-50 Mobility Controller Virtual Appliance on Pacstar PS451-1258 Series	Intel Xeon E-2254ML (CoffeeLake)	None

Table 5 – Vender Affirmed Operational Environments

#	Operating System	Hardware / Virtual Platform
1	ArubaOS 8.10	70xx Mobility Controllers
2	ArubaOS 8.10	72xx Mobility Controllers
3	ArubaOS 8.10	90xx Gateways
4	ArubaOS 8.10	92xx Gateways
5	ArubaOS 8.10	AP-51x and AP-57x Wireless Access Points
6	ArubaOS 8.10	AP-50x and AP-56x Wireless Access Points
7	ArubaOS 8.10	AP-53x, AP-55x, AP-58x, and AP-63x Wireless Access Points
8	ArubaOS 8.10	MCR-HW-xxx Mobility Conductor Hardware Appliances
9	ArubaOS 8.10 on VMWare ESXi 7.0	MC-VA-xxx Mobility Controller Virtual Appliances on HPE ProLiant ML110 Gen10
10	ArubaOS 8.10 on VMWare ESXi 7.0	MCR-VA-xxx Mobility Conductor Virtual Appliances on HPE ProLiant ML110 Gen10
11	ArubaOS 8.10 on VMWare ESXi 7.0	Virtual Appliances on HPE EdgeLine 20
12	ArubaOS 8.10 on VMWare ESXi 7.0	Virtual Appliances on PacStar PS451-1258 Series
13	ArubaOS 8.10 on VMWare ESXi 7.0	Virtual Appliances on device running an equivalent Intel processor (Intel Atom, i5, i7, or Xeon)

2.4 Excluded Components

There are no excluded components for the module.

2.5 Modes of Operation

The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware) is one of the Hewlett Packard Enterprise cryptographic modules that provide cryptographic services for the host ArubaOS operating system, and is installed automatically when a trusted and verified ArubaOS is booted on an Hewlett Packard Enterprise host device.

2.5.1 Approved Mode

When the module starts up successfully, after passing all the Cryptographic Algorithm Self-Tests (CASTs) and Pre-Operational Self-Tests (POSTs), and following the guidance in section 11.1, Start-up Procedures, the module is operating in the Approved mode of operation, provided that the guidelines on services, algorithms, and key management found in this Security Policy are followed.

2.5.2 Non-Approved Mode

When the module starts up but FIPS Settings are not enabled as per the guidance in section 11.1, Start-up Procedures, then the module is operating in non-Approved mode of operation.

2.6 Approved Algorithms

The firmware in the Hewlett Packard Enterprise *Aruba OpenSSL Module* contains the following cryptographic algorithm implementations that will be used for the corresponding security services supported by the module in the Approved mode.

Table 6 – Approved Algorithms

CAVP Cert.	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2690	AES [FIPS 197] [SP 800-38A]	CBC, CFB128, ECB, CTR (256, ext only, encryption only)	128, 192, 256	Data Encryption/Decryption
A2690	AES [FIPS 197] [SP 800-38C] [SP 800-38D] [SP 800-38F]	CCM, GCM, KW	128, 256	Data Encryption/Decryption
Vendor Affirmed[1]	CKG [SP 800-133 Rev2]	CTR_DRBG	N/A	Cryptographic Key Generation (using output from DRBG[2] as per IG D.H)
A2690	CVL IKEv1, TLS, SSH, SNMP[3] [SP 800-135 Rev1]	IKEv1: DSA, PSK TLS: v1.0/1.1	IKEv1: DH 2048-bit; SHA2-256, SHA2-384 TLS: SHA2-256, SHA2-384, SHA2-512 SSH: SHA-1	Key Derivation
A2690	CVL TLS[4] [SP 800-135 Rev1]	TLS: v1.2	TLS: SHA2-256, SHA2-384, SHA2-512	Key Derivation (using TLS v1.2 extended master secret as per IG D.Q and RFC 7627)
A2690	DRBG[5] [SP 800-90A Rev1]	AES CTR	256	Deterministic Random Bit Generation
A2690	DSA[6] [FIPS 186-4]	keyGen, pqgGen	L=2048, N=256, SHA2-256	Key and Parameter Generation
A2690	ECDSA [FIPS 186-4]	KeyGen, KeyVer, SigGen, SigVer	KeyGen: P-256, P-384 KeyVer: P-256, P-384 SigGen: P-256, P-384 with SHA2-256, SHA2-384, SHA2-512 SigVer: P-256, P-384 with SHA-1, SHA2-256, SHA2-384, SHA2-512	Key Generation and Verification, Digital Signature Generation and Verification

[1] Vendor Affirmed algorithms are approved by the CMVP but CAVP testing is not yet available.

[2] Resulting symmetric keys and seeds used for asymmetric key generation are unmodified output from SP 800-90A Rev1 DRBG.

[3] This KDF is used in the Approved IKEv1, HTTP over TLS, EAP-TLS, SSH and SNMP services. No parts of the IKEv1, TLS, SSH and SNMP protocols, other than the KDF, have been reviewed or tested by the CAVP and CMVP.

[4] This KDF is used in the Approved HTTP over TLS and EAP-TLS services. No parts of the TLS protocols, other than the KDF, have been reviewed or tested by the CAVP and CMVP.

[5] Refer to section 9.1, Non-Deterministic Random Number Generation Specification for entropy source details.

[6] DSA was CAVP tested but is only used as a pre-requisite for DH.

A2690	HMAC [FIPS 198-1]	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	(minimum 112 bits)	Message Authentication
A2690	KBKDF [SP 800-108 Rev1]	CTR	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	Key-based Key Derivation
A2690	KAS-SSC [SP 800-56A Rev3]	FFC: dhEphem, ECC: Ephemeral Unified	FFC: FC with SHA2-256, MODP-2048 with SHA2-256 ECC: P-256 with SHA2-256, P-384 with SHA2-384 KAS Roles - initiator, responder	Key Agreement Scheme – Shared Secret Computation (as per IG D.F, Scenario 2 (2))
A2690	KDA [SP 800-56C Rev2]	Two-step key derivation	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	Key Derivation Algorithm
A2690	RSA [FIPS 186-2]	SigVer: SHA-1[7], SHA2-256, SHA2-384, SHA2-512 PKCS1 v1.5	1024 (for legacy SigVer only), 2048	Digital Signature Verification
A2690	RSA [FIPS 186-4]	KeyGen, SigGen: SHA2-256, SHA2-384, SHA2-512 PKCS1 v1.5 SigVer: SHA-1[8], SHA2-256, SHA2-384, SHA2-512 PKCS1 v1.5	KeyGen: 2048 SigGen: 2048 SigVer: 1024 (for legacy SigVer only), 2048	Key Generation, Digital Signature Generation and Verification
A2690	Safe Primes [SP 800-56A Rev3]	KeyGen, KeyVer	Safe Prime Groups: MODP-2048	Safe Primes Key Generation and Key Verification
A2690	SHS [FIPS 180-4]	SHA-1, SHA2-256, SHA2-384, SHA2-512 Byte Only	160, 256, 384, 512	Message Digest
AES A2690	KTS [SP 800-38F]	AES-GCM[9]	128, 256	Key Wrapping / Key Transport via IKE/IPSec
AES A2690 HMAC A2690	KTS [SP 800-38F] [FIPS 198-1]	AES-CBC[10] HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	128, 192, 256 (minimum 112 bits)	Key Wrapping / Key Transport via IKE/IPSec

[7] SHA-1 is only Approved for use with Signature Verification.

[8] SHA-1 is only Approved for use with Signature Verification.

[9] AES-GCM is an authenticated encryption algorithm that is approved for use in key transport per FIPS 140-3 IG D.G. This key establishment methodology provides 128 or 256 bits of encryption strength.

[10] AES-CBC combined with HMAC is approved for use in key transport per FIPS 140-3 IG D.G. This key establishment methodology provides between 128 and 256 bits of encryption strength.

2.7 Non-Approved Cryptographic Algorithms Allowed in the Approved Mode of Operation

The cryptographic module implements no non-Approved algorithms allowed for use in the Approved mode of operation.

2.8 Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed

The cryptographic module implements the following non-Approved algorithms allowed in the Approved mode of operation with no security claimed, as per I.G. 2.4.A.

Table 7 – Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed

Algorithm	Caveat	Use / Function
MD5 within TLS	[no security claimed]	Used within TLS 1.0/1.1 only

2.9 Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

The cryptographic module implements the following non-Approved algorithms that are not permitted for use in the Approved mode of operations.

Table 8 – Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

Algorithm	Use / Function
DES	Used for older versions of WEP in non-Approved mode
HMAC-MD5	Used for older versions of WEP in non-Approved mode
MD5	Used for older versions of WEP in non-Approved mode
RC4	Used for older versions of WEP in non-Approved mode
Null Encryption	Used for older versions of WEP in non-Approved mode
RSA	Non-compliant less than 112 bits, or when used with SHA-1 for signature generation, or when other than 2048-bit modulus sizes are used, or when RSA-based Key Transport using PKCS#1-v1.5 is selected
Diffie-Hellman	key agreement; non-compliant less than 112 bits of encryption strength
EC Diffie-Hellman	key agreement; non-compliant less than 112 bits of encryption strength
ECDSA	non-compliant when using 186-2 signature generation
Triple-DES-CBC	As used in IKE/IPSec

3 Cryptographic Module Interfaces

As a firmware module, the module interfaces are defined as Software or Firmware Module Interfaces (SFMI), and there are no physical ports. The logical interfaces are defined as the API of the cryptographic module. The interfaces are listed in the table below.

All data output via data output interface is inhibited when the module is performing pre-operational tests or zeroization or when the module enters error state.

Table 9 – Ports and Interfaces

Physical Port	Logical Interface	Data That Passes Over the Interface
N/A	Data Input	API input parameters for data
N/A	Data Output	API output parameters for data
N/A	Control Input	API function calls
N/A	Status Output	API return codes, status information, error messages
N/A	Power	None

Notes:

- Module API return codes to calling application: 1 = success, 0 = failure.
- The module does not implement a control output interface.

4 Roles, Services, and Authentication

The following section lists the roles supported by the module, authentication mechanisms used by the module, and services (both security and non-security, Approved and non-Approved) available from the module.

4.1 Authentication

The Hewlett Packard Enterprise *Aruba OpenSSL Module* does not provide any identification or authentication methods of its own. The CO and the User roles are implicitly identified by the service requested.

4.2 Roles

The module supports two distinct operator roles: the Crypto Officer role and the User role. These roles are implicitly assumed by the operator of the module when performing a service. The module does not support multiple concurrent operators, a maintenance role, nor bypass capability.

Table 10 – Roles and Authentication

Name	Authentication Methods	Authentication Strength
Crypto Officer	N/A – Authentication not required for Level 1	N/A
User	N/A – Authentication not required for Level 1	N/A

The table below lists descriptions of the services available to the roles, with input and output.

Table 11 – Roles, Service Commands, Input, Output

Role	Service	Input	Output
User	Data encryption, decryption	Plaintext or ciphertext, key	Ciphertext or plaintext
User	Key derivation function	Shared secrets, inputs (IKEv1, TLS, SSH, WPA2/WPA3)	Derived keys
User	Deterministic Random Bit Generation	Entropy input string, seed, internal state	Random number
User	Digital signature	RSA or ECDSA public and private keys	RSA or ECDSA digital signature generated or verified
User	Message authentication	Message, HMAC key	Message authentication code
User	Key agreement	DH (FFC), ECDH key agreement primitives	Shared secret, derived keys
User	Safe Primes key generation and verification	DH (FFC) domain parameters	DH (FFC) private key
User	Key pair generation	Key size or curve size	RSA, DSA (FFC), or ECDSA key pairs
User	Key wrapping / Key transport	AES key	Wrapped keys
User	Message digest	Message	Digest of the message
Crypto Officer	Zeroization	Command	Progress information
Crypto Officer	Status function	Commands and configuration data	Status of commands and configurations
User	Show Version	Command	Name and version of the module
Crypto Officer	Reboot Module	Command	Progress information
Crypto Officer	Self-Test triggered by Crypto Officer reboot	Module reboot	Progress information
Crypto Officer	Approved mode enable/disable	Command	Progress information

Note: The Crypto Officer must ensure that the module is kept in the Approved mode of operation, following the guidance in section 11.1, Start-up Procedures, and provided that the guidelines on services, algorithms, and key management found in this Security Policy are followed.

4.3 Services

The module provides various services depending on role. These are described in the sections below.

The meaning of the letters used to describe the ‘Access Rights to Keys and/or SSPs’ are:

- **G** – **Generate** The module generates or derives the Key/SSP.
- **R** – **Read** The Key/SSP is read from the module (e.g. the Key/SSP is output).
- **W** – **Write** The Key/SSP is updated, imported, or written to the module.
- **E** – **Execute** The module uses the Key/SSP in performing a cryptographic operation.
- **Z** – **Zeroize** The module zeroizes the Key/SSP.

4.3.1 Approved Services

See the tables below for descriptions of the services, Approved security functions, keys and/or SSPs available to the module’s roles.

The Hewlett Packard Enterprise *Aruba OpenSSL Module* is one of the components within the ArubaOS firmware package which runs on the host device. ArubaOS includes CLI commands, some of which interact with the module via APIs. Successful completion of a security service (via API return code for success) when the module is in Approved mode (see section 11.1, Start-up Procedures) denotes use of approved security service.

Table 12 – Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs [row # in SSPs/Keys Used table]	Roles	Access Rights to Keys and/or SSPs	Indicator
Data encryption, decryption	Encrypt or decrypt data	AES-CBC AES-GCM AES-ECB AES-CTR AES-CFB128 AES-CCM AES-KW (A2690)	[19] IKE Session Encryption Key [20] IPSec Session Encryption Key [22] SSHv2 Session Key [26] TLS Session Encryption Key [29] SNMPv3 Authentication Key [31] SNMPv3 Privacy Key [36] WPA2/WPA3 Session Key [38] WPA2/WPA3 Group Transient Key (GTK)	User	W/E W/E W/E W/E W/E W/E W/E W/E	API return code for success
Deterministic Random Bit Generation	Generate random numbers with SP800-90A Rev1 Approved AES-256 CTR_DRBG	CTR_DRBG (A2690)	[1] DRBG Entropy Input [2] DRBG Seed [3] DRBG Key [4] DRBG V	User	W/E G/E G/E G/E	API return code for success
Digital signature	Generate or verify RSA or ECDSA digital signatures	CTR_DRBG RSA SigGen RSA SigVer ECDSA SigGen ECDSA SigVer (A2690)	[1] DRBG Entropy Input [3] DRBG Key [4] DRBG V [11] RSA Private Key [12] RSA Public Key [13] ECDSA Private Key [14] ECDSA Public Key	User	W/E E E W/E W/E W/E W/E W/E	API return code for success

Safe Primes key generation and verification	Diffie-Hellman key generation and verification using safe primes	CTR_DRBG Safe Primes KeyGen Safe Primes KeyVer (A2690)	[1] DRBG Entropy Input [3] DRBG Key [4] DRBG V [5] DH Private Key [6] DH Public Key	User	W/E E E W/E W/E	API return code for success
Key derivation function	Key derivation through SP800-135rev1-KDF (IKEv1-KDF, TLS-KDF, TLSv1.2-KDF-RFC7627, SSHv2-KDF, SNMPv3-KDF), SP800-108rev1-KDF (KBKDF) and SP800-56Crev2	KBKDF KDA Two Step IKEv1-KDF TLS-v1.0/1.1-KDF TLS-v1.2-KDF-RFC7627 SSHv2-KDF SNMPv3-KDF (A2690)	[7] DH Shared Secret [10] ECDH Shared Secret [15] IKE Pre-Shared Key [16] skeyid [17] skeyid_d [18] IKE Session Authentication Key [19] IKE Session Encryption Key [20] IPSec Session Encryption Key [21] IPSec Session Authentication Key [22] SSHv2 Session Key [23] SSHv2 Session Authentication Key [24] TLS Pre-Master Secret [25] TLS Master Secret [26] TLS Session Encryption Key [27] TLS Session Authentication Key [28] SNMPv3 Authentication Password [29] SNMPv3 Authentication Key [30] SNMPv3 Engine ID [31] SNMPv3 Privacy Key [32] SNMPv3 Privacy Protocol Password [33] WPA2/WPA3 Pre-Shared Secret [34] WPA2/WPA3 Pair-Wise Master Key (PMK) [35] WPA2/WPA3 PairWise Transient Key (PTK) [36] WPA2/WPA3 Session Key [37] WPA2/WPA3 Group Master Key (GMK) [38] WPA2/WPA3 Group Transient Key (GTK)	User	W/E W/E W/E G/Z G/R G/R G/R G/R G/R G/R G/R W/E W/E G/R G/R W/E G/R W/E G/R W/E W/E G/R G/R G/R G/R G/R	API return code for success
Message authentication	Generate or verify data integrity with HMAC key	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512 (A2690)	[18] IKE Session Authentication Key [21] IPSec Session Authentication Key [23] SSHv2 Session Authentication Key [27] TLS Session Authentication Key [35] WPA2/WPA3 PairWise Transient Key (PTK)	User	W/E W/E W/E W/E W/E	API return code for success
Key agreement	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module)	KAS-ECC-SSC KAS-FFC-SSC (A2690)	[5] DH Private Key [6] DH Public Key [7] DH Shared Secret [8] ECDH Private Key [9] ECDH Public Key [10] ECDH Shared Secret	User	W/E W/E G/R W/E W/E G/R	API return code for success

Key wrapping / Key transport	AES key wrapping	AES-GCM AES-KW AES-CBC with HMAC (A2690)	[19] IKE Session Encryption Key [20] IPSec Session Encryption Key	User	W/E W/E	API return code for success
Key pair generation	Generate RSA, FFC, or ECDSA key pairs	CTR_DRBG DSA/FFC keyGen DSA/FFC pqgGen Safe Primes KeyGen Safe Primes KeyVer ECDSA/ECC KeyGen ECDSA/ECC KeyVer RSA KeyGen RSA KeyVer (A2690)	[1] DRBG Entropy Input [3] DRBG Key [4] DRBG V [5] DH Private Key [6] DH Public Key [8] ECDH Private Key [9] ECDH Public Key [11] RSA Private Key [12] RSA Public Key [13] ECDSA Private Key [14] ECDSA Public Key	User	W/E E E G/R G/R G/R G/R G/R G/R G/R G/R	API return code for success
Message digest	Generate a SHA-1 or SHA2 message digest	SHA-1 SHA2-256 SHA2-384 SHA2-512	None	User	None	API return code for success

Table 13 – Approved Services Not Using Any Approved Security Functions

Service	Description	Approved Security Functions	Keys and/or SSPs [row # in SSPs/Keys Used table]	Roles	Access Rights to Keys and/or SSPs	Indicator
Approved mode enable/disable	The Crypto Officer enables the Approved mode by following the procedures under section 11.1, Start-up Procedures, to ensure the Aruba host device is configured for Secure Operations.	None	None	Crypto Officer	None	API return code for success
Status Function	Crypto Officer may use ArubaOS CLI "show" commands on host to view logs and status. Please see ArubaOS CLI guide for details.	None	None	Crypto Officer (N/A for system status via host LEDs)	None	API return code for success
Show Version	User may use ArubaOS CLI "`show ver`" command on host to view module name and version in log.	None	None	User	None	API return code for success
Reboot module	The Crypto Officer may remotely trigger a reboot. The module can also reboot by removing/replacing power to the host device.	None	None	Crypto Officer	None	API return code for success
Self-Test triggered by Crypto Officer reboot	Perform FIPS pre-operational self-tests and conditional cryptographic algorithm tests (refer to section 10, Self-Tests) on demand through module reboot.	None	None	Crypto Officer	None	API return code for success

Zeroization	The cryptographic keys stored in SDRAM memory can be zeroized by rebooting the module. The cryptographic keys (IKE Pre-shared key and RSA/ECDSA Private/Public keys) stored in the host flash can be zeroized by using the ArubaOS command '`wipe out flash`' on the host or by overwriting with a new secret. Please see ArubaOS CLI guide for details.	None	All SSPs will be destroyed.	Crypto Officer	Z	API return code for success

4.3.2 Non-Approved Services

The non-Approved services listed in the table below are available in the non-Approved mode but are not available in the Approved mode (see section 11.1, Start-up Procedures). The module does not support a degraded mode of operation.

Table 14 – Non-Approved Services

Service	Description	Algorithms Accessed	Roles
IPSec/IKE using Triple-DES	IPSec/IKE key management using Triple-DES. This is a non-Approved service not available in the Approved mode.	Triple-DES	User
Use of non-Approved algorithms and/or sizes.	If the module has not been provisioned to operate in the Approved mode, then non-Approved algorithms and/or sizes are available for use – see above Table 9, Non-Approved Algorithms Not Allowed in the Approved Mode of Operation. This is a non-Approved service not available in the Approved mode.	Non-Approved algorithms and/or sizes	User
RSA-based Key Transport using PKCS#1-v1.5	RSA-based key transport using RSA PKCS#1-1.5 and an RSA modulus at least 2048 bits long, as per section 8.1 of RFC 2313.	RSA PKCS#1-v1.5	User

Note:

- Log message displayed each time RSA-based key transport using RSA PKCS#1-1.5 is executed:
  ```
  This operation is calling RSA PKCS#1-1.5 for key transport, non-approved under FIPS 140-3.
  Please ensure a FIPS 140-3 compliant transport scheme is configured
  ```
- For additional information, please refer to the *ArubaOS 8.10 User Guide*.

5 Software / Firmware Security

The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware version 1.0) is an Hewlett Packard Enterprise cryptographic module that provides cryptographic services for the ArubaOS operating system. The module is one of the components within the ArubaOS firmware package and is installed automatically when a trusted and verified ArubaOS firmware package signed by Hewlett Packard Enterprise is booted on an Hewlett Packard Enterprise host device.

The module performs a firmware integrity test when powered on (refer to Self-Tests for details). All cryptographic algorithm self-tests are run at power-up, prior to the first operational use of the cryptographic algorithm. The firmware integrity test verifies the integrity of the module by comparing a calculated HMAC-SHA-1 value against the stored HMAC value.

The operator can initiate the firmware integrity test on demand by rebooting the host device. Rebooting also zeroizes all SSPs stored in SDRAM memory. All data output via the data output interface is inhibited until the firmware integrity test has completed successfully. If the firmware integrity test fails, the module enters the error state (while in this state, the module provides no functionality). The temporary values generated during the firmware integrity test are zeroized upon completion of the integrity test. After the ArubaOS firmware boot, the operator can determine the version of the loaded module through reviewing the log and by using the show version ArubaOS CLI command on the host (use the link in the section Full Documentation to refer to *ArubaOS 8.10 Command-Line Interface Reference Guide* and *ArubaOS 8.10 User Guide*).

6 Operational Environment

The operational environment is limited.

The control plane Operating System (OS) is Linux, a multi-threaded operating system that supports memory protection between processes. Access to the underlying Linux implementation is not provided directly. Only Hewlett Packard Enterprise provided interfaces are used. The Hewlett Packard Enterprise *Aruba OpenSSL Module* is one of the components within the ArubaOS firmware package which runs on the host device. ArubaOS includes Command Line Interface (CLI) commands, some of which interact with the module via APIs. The ArubaOS CLI and the module APIs are restricted command sets. These operating control mechanisms protect against unauthorized execution, unauthorized modification, and unauthorized reading of SSPs, control and status data.

The module was tested on the platforms listed above in section 2.3, Table 4, Tested Operational Environments.

7 Physical Security

The Hewlett Packard Enterprise *Aruba OpenSSL Module* is a firmware type module and obtains its physical security from the host platform. As per FIPS 140-3 for multiple-chip standalone cryptographic modules at Security Level 1, the host platform consists of production-grade components within a production-grade enclosure. The platforms listed above in section 2.3 meet these requirements.

8 Non-Invasive Security

Since the module has not been purposely designed, built and publicly documented to include non-invasive mitigation techniques, the Non-Invasive Security requirements are not applicable.

9 Sensitive Security Parameter (SSP) Management

The following are the Sensitive Security Parameters (SSPs) used in the module. As specified in the Zeroization column of the following table, the majority of SSPs/Keys used in the module are zeroized implicitly by rebooting the module, indicated implicitly via the successful completion of the module reboot service. The Hewlett Packard Enterprise *Aruba OpenSSL Module* is one of the components within the ArubaOS firmware package which runs on the host device, thus the module is rebooted when the host device is rebooted. ArubaOS includes CLI commands. As specified in the Zeroization column of the following table, there are a minority of SSPs/Keys used in the module that are stored in the host flash. The host flash can be zeroized explicitly by using the ArubaOS CLI command '`wipe out flash`' on the host device.

Table 15 – SSPs/Keys Used in the Module

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
General Keys/SSPs									
1	DRBG Entropy Input – CSP	512 bits	SP800-90A Rev1 CTR_DRBG AES-256 Cert. #A2690	64 bytes are retrieved from the entropy source read from /dev/random on each call by any service that requires a random number.	Import: From Aruba CPU Jitter Entropy Source Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Entropy inputs to the DRBG function, used to construct the DRBG Seed.
2	DRBG Seed – CSP	384 bits	SP800-90A Rev1 CTR_DRBG AES-256 Cert. #A2690	Generated using DRBG derivation function that includes the entropy input from the entropy source read from /dev/random.	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Input to the DRBG that determines the internal state of the DRBG (DRBG Key and V).
3	DRBG Key – CSP	256 bits	SP800-90A Rev1 CTR_DRBG AES-256 Cert. #A2690	Derived from the DRBG Seed.	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	This is the DRBG key used for SP800-90A Rev1 CTR_DRBG during generation of random numbers.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
4	DRBG V – CSP	128 bits	SP800-90A Rev1 CTR_DRBG AES-256 Cert. #A2690	Derived from the DRBG Seed.	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Internal V value used as part of SP800-90A Rev1 CTR_DRBG during generation of random numbers.
5	Diffie-Hellman Private Key – CSP	112 bits	Diffie-Hellman Group 14 Cert. #A2690	Generated internally in compliance with Diffie-Hellman key agreement scheme by calling Approved DRBG (Cert. #A2690)	Import: From calling application Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used during the IPSec handshake to establish the Diffie-Hellman Shared Secret.
6	Diffie-Hellman Public Key – PSP	112 bits	Diffie-Hellman Group 14 Cert. #A2690	Generated internally in compliance with Diffie-Hellman key agreement scheme by calling Approved DRBG (Cert. #A2690)	Import: From calling application Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used during the IPSec handshake to establish the Diffie-Hellman Shared Secret.
7	Diffie-Hellman Shared Secret – CSP	112 bits	Diffie-Hellman Group 14 Cert. #A2690	N/A	Import: N/A Export: To calling application	Established during Diffie-Hellman Exchange.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for deriving IPSec/IKE and SSH cryptographic keys.
8	EC Diffie-Hellman Private Key – CSP	Curves: P-256 or P-384	EC Diffie-Hellman Cert. #A2690	Generated internally by calling Approved DRBG (Cert. #A2690) during EC Diffie-Hellman Exchange.	Import: From calling application Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for establishing EC Diffie-Hellman Shared Secret.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
9	EC Diffie-Hellman Public Key – PSP	Curves: P-256 or P-384	EC Diffie-Hellman Cert. #A2690	Generated internally by calling Approved DRBG (Cert. #A2690) during EC Diffie-Hellman Exchange.	Import: From calling application Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for establishing EC Diffie-Hellman Shared Secret.
10	EC Diffie-Hellman Shared Secret – CSP	Curves: P-256 or P-384	EC Diffie-Hellman Cert. #A2690	N/A	Import: N/A Export: To calling application	Established during EC Diffie-Hellman Exchange.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for deriving IPSec/IKE and TLS cryptographic keys.
11	RSA Private Key – CSP	128 bits	RSA Private Key Cert. #A2690	Generated by calling Approved DRBG (Cert. #A2690) in the module, in compliance with FIPS 186-4 RSA key pair generation method.	Import: From calling application Export: To calling application	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command '`wipe out flash`' on host.	Used for IKEv1, TLS, Online Certificate Status Protocol (OCSP) (signing OCSP messages) and Extensible Authentication Protocol (EAP) -TLS peers authentication.
12	RSA Public Key – PSP	128 bits	RSA Public Key Cert. #A2690	Generated by calling Approved DRBG (Cert. #A2690) in the module, in compliance with FIPS 186-4 RSA key pair generation method.	Import: From calling application Export: To calling application	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command '`wipe out flash`' on host.	Used for IKEv1, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
13	ECDSA Private Key – CSP	Curves: P-256 or P-384	ECDSA SigGen Cert. #A2690	Generated by calling Approved DRBG (Cert. #A2690) in the module, in compliance with FIPS 186-4 ECDSA key pair generation method.	Import: From calling application Export: To calling application	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command ‘`wipe out flash`’ on host.	Used for IKEv1, TLS and EAP-TLS peers authentication.
14	ECDSA Public Key – PSP	Curves: P-256 or P-384	ECDSA SigGen Cert. #A2690	Generated by calling Approved DRBG (Cert. #A2690) in the module, in compliance with FIPS 186-4 ECDSA key pair generation method.	Import: From calling application Export: To calling application	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command ‘`wipe out flash`’ on host.	Used for IKEv1, TLS and EAP-TLS peers authentication.
IPSec/IKE									
15	IKE Pre-Shared Key – CSP	8 - 64 ASCII or 64 HEX characters	Shared Secret Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command ‘`wipe out flash`’ on host or by overwriting with a new secret.	Used for IKEv1 peers authentication.
16	skeyid – CSP	160 / 256 / 384 bits	Shared Secret Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module.	A shared secret known only to IKEv1 peers. Used for deriving other keys in IKEv1 protocol implementation.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
17	skeyid_d – CSP	160 / 256 / 384 bits	Shared Secret Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module.	A shared secret known only to IKEv1 peers. Used for deriving IKEv1 Session Authentication Key.
18	IKE Session Authentication Key – CSP	160 / 256 / 384 bits	HMAC-SHA-1/ HMAC-SHA2-256/384 Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	The IKE session (IKE Phase I) authentication key. Used for IKEv1 payload integrity verification.
19	IKE Session Encryption Key – CSP	128 / 192 / 256 bits	AES (CBC) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	The IKE session (IKE Phase I) encrypt key. Used for IKE payload protection.
20	IPSec Session Encryption Key – CSP	128 / 192 / 256 bits 128 / 256 bits	AES (CBC) and AES (GCM) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	The IPSec (IKE phase II) encryption key. Used for IPSec traffics protection. IPSec session encryption keys can also be used for the Double Encrypt feature.
21	IPSec Session Authentication Key – CSP	160 bits	HMAC-SHA-1 Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (IKEv1).	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	The IPSec (IKE Phase II) authentication key. Used for IPSec traffics integrity verification.
SSHv2									
22	SSHv2 Session Key – CSP	128 / 192 / 256 bits	AES (CBC) and AES (CTR) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (SSHv2).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for SSHv2 traffics protection.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
23	SSHv2 Session Authentication Key – CSP	160 bits	HMAC-SHA-1 / HMAC-SHA1-96 Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (SSHv2).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for SSHv2 traffics protection.
TLS									
24	TLS Pre-Master Secret – CSP	112 to 8192 bits	Shared Secret Cert. #A2690	N/A	Import: From calling application Export: N/A	DH/ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	This key is generated by the client and sent across to the server (controller), encrypted using the public key provided by the server. Used for TLS traffics authentication.
25	TLS Master Secret – CSP	384 bits	Shared Secret Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (TLS). If TLS v1.2, then uses TLS v1.2 version of KDF that supports RFC7627.	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for TLS traffics protection.
26	TLS Session Encryption Key – CSP	128 / 256 bits	AES (CBC) and AES (GCM) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (TLS).	Import: N/A Export: N/A	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for TLS traffics protection.
27	TLS Session Authentication Key – CSP	160 / 256 / 384 bits	HMAC-SHA-1/ HMAC-SHA2-256/384 Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (TLS).	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for TLS traffics integrity verification.
SNMPv3									

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
28	SNMPv3 Authentication Password – CSP	8-31 ASCII characters	Password Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command '`wipe out flash`' on host or by overwriting with a new secret.	Used for SNMPv3 authentication.
29	SNMPv3 Authentication Key – CSP	128 bits	AES (CFB) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (SNMPv3).	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for SNMPv3 authentication.
30	SNMPv3 Engine ID – CSP	10-24 HEX characters	Password Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command '`wipe out flash`' on host or by overwriting with a new secret.	A unique string used to identify the SNMP engine.
31	SNMPv3 Privacy Key – CSP	128 bits	AES (CFB) Cert. #A2690	Derived via key derivation function defined in SP800-135 Rev1 KDF (SNMPv3).	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used for SNMPv3 traffics protection.
32	SNMPv3 Privacy Protocol Password – CSP	8-31 ASCII characters	Password Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command '`wipe out flash`' on host or by overwriting with a new secret.	A unique string used to protect SNMP privacy protocol.
WPA2/WPA3									

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
33	WPA2/WPA3 Pre-Shared Secret – CSP	8-63 ASCII or 64 HEX characters	Shared Secret Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in host Flash memory (plaintext).	Zeroized by using ArubaOS command 'wipe out flash' on host or by overwriting with a new secret.	Used for WPA2/WPA3 client/server authentication.
34	WPA2/WPA3 Pair-Wise Master Key (PMK) – CSP	256 bits	Shared Secret Cert. #A2690	N/A	Import: From calling application Export: N/A	N/A	Stored in SDRAM (plaintext).	Zeroized by rebooting the module.	Used to derive the Pairwise Transient Key (PTK) for WPA2/WPA3 communications.
35	WPA2/WPA3 Pairwise Transient Key (PTK) – CSP	384 bits	HMAC-SHA-1 Cert. #A2690	Derived via key derivation function defined in SP800-108 Rev1 and SP800-56C Rev2.	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used to derive the WPA2/WPA3 Session Key.
36	WPA2/WPA3 Session Key – CSP	128 bits, 128 / 256 bits	AES (CCM) and AES (GCM) (WPA3 only) Cert. #A2690	Derived during WPA2/WPA3 4-way handshake by using the KDF defined in SP800-108 Rev1 and SP800-56C Rev2.	Import: From calling application Export: To calling application	N/A	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.	Used as the WPA2/WPA3 Session Key.
37	WPA2/WPA3 Group Master Key (GMK) – CSP	256 bits	Shared Secret Cert. #A2690	Generated internally by calling Approved DRBG (Cert. #A2690).	Import: N/A Export: To calling application	N/A	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module	Used to derive WPA2/WPA3 Group Transient Key GTK.

#	Key / SSP Name / Type	Security Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use and Related Keys
38	WPA2/WPA3 Group Transient Key (GTK) – CSP	256 bits	AES (CCM) and AES (GCM) Cert. #A2690	N/A	Import: N/A Export: To calling application	Derived from WPA2/WPA3 GMK by using the KDF defined in SP800-108 Rev1 and SP800-56C Rev2.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module	The GTK is the WPA2/WPA3 session key used for broadcast communications protection.

Notes:

- AES GCM IV generation is performed in compliance with the Implementation Guidance C.H scenario 1 for TLS.
  - For TLS, the module is compliant with RFC 5288. Specifically, the module uses RFC 5288 compliant TLS 1.2 GCM Cipher Suites (TLS_ECDHE_RSA and TLS_ECDHE_ECDSA with AES_128_GCM_SHA256 and AES_256_GCM_SHA384) for TLS as per NIST SP 800-52 Rev2 section 3.3.1.
  - When the nonce_explicit part of the IV for TLS exhausts the maximum number of possible values for session key for TLS, either party to the client/server for TLS that encounters this condition triggers a handshake with TLS to establish a new encryption key.
- AES GCM IV generation is performed in compliance with the Implementation Guidance C.H scenario 4 for WPA3. The session is reauthenticated by the module after 24 hours which resets the AES GCM IV counter. The 24 hours (86400 seconds) interval is the default setting and shall not be changed.
- AES GCM IV is generated internally in compliance with the Implementation Guidance C.H scenario 3 and SP800-38D section 8.2.2.
- In case the module's power is lost and then restored, a new key for use with the AES-GCM encryption/decryption shall be established.
- CKG (vendor affirmed to SP 800-133 Rev2): For keys identified as being "Generated internally", the generated seed used in the asymmetric key generation is an unmodified output from the Approved DRBG (Cert. #A2690).
- The Approved DRBG (Cert. #A2690) generates a minimum of 256 bits of entropy for use in key generation.
- Sensitive Security Parameters (SSPs) can be Critical Security Parameters (CSPs) or Public Security Parameters (PSPs).
- Keys established while operating in the non-Approved mode cannot be used in the Approved mode, and vice versa.

9.1 Non-Deterministic Random Number Generation Specification

Table 16 – Non-Deterministic Random Number Generation Specification

Entropy Sources	Minimum Number of Bits of Entropy	Details
Aruba CPU Jitter Entropy Source (see NIST Entropy Source Validation (ESV) program certificate E7)	Oversampling of 512 bits is performed to ensure that 256 bits of entropy is available to the DRBG.	The module employs a SP800-90Arev1-compliant Deterministic Random Bit Generator (DRBG) using an AES-256 CTR_DRBG mechanism with Derivation Function (DF) for random number generation (Cert. #A2690). The employed DRBG uses a SP800-90B-compliant non-physical entropy source that uses CPU jitter provided by the operational environment as a noise source (Jitterentropy (JENT) with SHA-3 as the vetted conditioning component) which is within the module host's physical boundary but outside of the module's cryptographic boundary. The entropy source performs the SP800-90B Section 4.4 Approved Continuous Health Tests (RCT and APT).

10 Self-Tests

The module performs when powered on the Cryptographic Algorithm Self-Tests (CASTs) and Pre-Operational Self-Tests (POSTs). While the module is executing the cryptographic algorithm and pre-operational self-tests, services are not available, and input and output are inhibited. In addition, the module also performs Conditional self-tests. All cryptographic algorithm self-tests are run when the module is powered on, prior to the first operational use of the cryptographic algorithm. After the cryptographic algorithm, pre-operational, and conditional self-tests are passed successfully, the module transitions to the operational state.

When a cryptographic algorithm self-test or pre-operational self-test fails, or when a conditional self-test fails, the module enters the Critical Error state (while in this state, the module provides no functionality and inhibits data output), logs the error, and reboots automatically.

The Hewlett Packard Enterprise *Aruba OpenSSL Module* performs the following **Pre-Operational Self-Tests (POSTs)**:

Table 17 – Pre-Operational Self-Tests

Algorithm	Test Properties	Type	Details
Firmware Integrity Test	HMAC-SHA-1 with 128-bit key	KAT	The Firmware Integrity Test verifies the integrity of the module by comparing a calculated HMAC-SHA-1 value against the stored HMAC value from each shipped .hmac file. The KAT for the HMAC-SHA-1 is executed before the Firmware Integrity Test.

The Hewlett Packard Enterprise *Aruba OpenSSL Module* performs the following **Conditional Tests**:

Table 18 – Conditional Cryptographic Algorithm Tests

Algorithm	Test Properties	Type	Details	Condition
AES ECB	AES-ECB-128	KAT	Encrypt, Decrypt	Each run when module powered on, which is prior to the first operational use of the cryptographic algorithms
AES CCM	AES-CCM-192	KAT	Encrypt, Decrypt	
AES GCM	AES-GCM-256	KAT	Encrypt, Decrypt	
DRBG	AES-CTR-256, CTR_DRBG with DF, with and without PR	KAT	SP800-90A Rev1 Section 11.3 Health Tests for CTR_DRBG (Instantiate, Generate and Reseed)	
ECDSA	P-256, P-384	KAT	Sign, Verify	

Algorithm	Test Properties	Type	Details	Condition
HMAC	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	KAT		Each run when module powered on, which is prior to the first operational use of the cryptographic algorithms
KAS-SSC-ECC	Primitive 'Z' computation with P-256 curve	KAT	Ephemeral Unified SP 800-56A Rev3 based	
KAS-SSC-FFC	Shared secret computation, p=2048, q=256	KAT	dhEphem SP 800-56A Rev3 based	
KDA	Two-step KDF: HMAC-SHA-1, L=2048	KAT	SP 800-56C Rev2 based	
KBKDF	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	KAT	SP 800-108 Rev1 based key derivation	
KDF135	HMAC-SHA-1	KAT	SP 800-135 Rev1 based key derivation: IKEv1, TLS, SSH, SNMP	
RSA	2048, PKCS#1-v1.5	KAT	Sign, Verify	
SHS	SHA-1, SHA-256, SHA-384, SHA-512	KAT		

Table 19 – Conditional Pairwise Consistency Tests

Algorithm	Test Properties	Type	Details	Condition
ECC key pairs	P-256, P-384	PCT	Sign, Verify	
FFC key pairs	DH key pair generation	PCT	SP800-56A Rev3 assurances as per SP 800-56A Rev3 Section 5.6.2.1.4 for PCT	
RSA key pairs	2048, PKCS#1-v1.5	PCT	Sign, Verify	

Self-Test Types:

KAT = Known Answer Test, **PCT** = Pairwise Consistency Test

To see the results of the self-tests run by the module, use the ArubaOS CLI command on the host device:

```
show log crypto all
```

Upon successful completion of the power-up self-tests, the module displays results on the host device console:

```
Completed OpenSSL FIPS KAT test successfully.
```

In the event any self-test fails, the module will enter a Critical Error state (while in this state, the module provides no functionality and inhibits data output), logs the error, and reboots automatically. After a self-test failure, the module displays on the host device console:

```
FIPS POST: FAIL
Rebooting…
```

When the firmware integrity test fails at power-up, the module enters the Critical Error state, where the invalid host ArubaOS firmware file is deleted to clear the error. The host device will automatically reboot from the backup ArubaOS image on the secondary partition. The module displays on the host device console:

```
FATAL FIPS SELFTEST FAILURE
Rebooting…
```

11 Life-Cycle Assurance

The Hewlett Packard Enterprise *Aruba OpenSSL Module* is a firmware type module, and must run on an Hewlett Packard Enterprise hardware unit (e.g., Controller, Gateway, Conductor, or Access Point) or virtual appliance (e.g., VMWare ESXi or open source KVM hypervisor running on a hardware server unit (e.g., HPE ProLiant ML110 Gen10 or HPE EdgeLine 20)).

ArubaOS is the operating system for Hewlett Packard Enterprise Mobility Conductors, Mobility Controllers/Gateways, and controller-managed Hewlett Packard Enterprise Access Points (APs). The Hewlett Packard Enterprise *Aruba OpenSSL Module* (firmware) is one of the Hewlett Packard Enterprise cryptographic modules that provide cryptographic services for the host ArubaOS operating system running on the Hewlett Packard Enterprise hardware-based equipment or Hewlett Packard Enterprise virtual appliances.

11.1 Start-up Procedures

The Hewlett Packard Enterprise *Aruba OpenSSL Module* is one of the components within the ArubaOS firmware package in electronic form and is installed automatically when a trusted and verified ArubaOS is booted on an Hewlett Packard Enterprise host device. ArubaOS firmware in electronic form is installed by Hewlett Packard Enterprise technical support personnel or downloaded from the HPE Networking Support Portal (NSP) by authenticated licensed customer personnel.

11.1.1 Setting Up the Hewlett Packard Enterprise Controller, Gateway, Conductor, or Controller-managed Access Point (AP) and Running Hewlett Packard Enterprise Aruba OpenSSL Module Automatically

The Crypto Officer shall perform the following steps to set-up the Hewlett Packard Enterprise Controller, Gateway, Conductor, or controller-managed Access Point (AP) either as a host hardware unit or a virtual appliance:

1. Since the Hewlett Packard Enterprise *Aruba OpenSSL Module* firmware is a component of ArubaOS and is installed automatically when a trusted and verified ArubaOS firmware image is booted successfully on the Hewlett Packard Enterprise host device, the Crypto Officer (CO) shall review the *ArubaOS 8.10 Getting Started Guide, ArubaOS 8.10.0.x AP Software Quick Start Guide,* and *ArubaOS 8.10 Virtual Appliance Installation Guide*. Select the Hewlett Packard Enterprise host device running ArubaOS deployment scenario that best fits your installation and follow the scenario's deployment procedures.
2. Connect your PC or workstation to a line port (or virtual port mapped to the module interface) on the Hewlett Packard Enterprise Controller, Gateway, Conductor, or controller-managed Access Point (AP).
3. Enable power to the Hewlett Packard Enterprise Controller, Gateway, Conductor, or controller-managed Access Point (AP).
4. Monitor the Hewlett Packard Enterprise host device boot progress messages on the console.
5. Once ArubaOS is loaded successfully and operating normally on the Hewlett Packard Enterprise host device, check the console messages to make sure that all the power-up self-tests passed.
    a. Check that the following console message is displayed:
    ```
    Completed OpenSSL FIPS KAT test successfully.
    ```
    b. As specified in the Self-Tests section, if any of the checks fail, error messages will be displayed on the console. If the errors persist after the Hewlett Packard Enterprise device is rebooted, contact Hewlett Packard Enterprise.
6. Enable the Approved mode with the ArubaOS CLI on the host.
    a. Login to the Hewlett Packard Enterprise Controller, Gateway, or Conductor following the guidance from step 1. above (a controller-managed Access Point (AP) is placed in the Approved mode using a Staging Controller in the Approved mode as per the *Aruba AP Software Quick Start Guide)*.

b. Enable the Approved mode using the following ArubaOS CLI commands on the host:

```
#configure terminal
Enter Configuration commands, one per line. End with CNTL/Z
(config) #fips enable
(config) #exit
#write memory
Saving Configuration...
Configuration Saved.
```

c. To verify the Approved mode has been enabled, issue the ArubaOS CLI command on the host:

```
show fips     to see:     FIPS Settings:
                          Mode Enabled
```

11.2 Full Documentation

Documentation for any Hewlett Packard Enterprise product can be found on the HPE Networking Support Portal (NSP). Filters can be used to limit the displayed results by Product(s), Product Series, Version(s), and File Category.

For example,

- Full ArubaOS version 8.10 documentation for Hewlett Packard Enterprise Mobility Controllers, Virtual Mobility Controllers, Gateways, Mobility Conductors, and Access Points can be found at the link provided below after authentication.

  https://networkingsupport.hpe.com/downloads;pageSize=100;fileTypes=DOCUMENT;products=Aruba%20Access%20Points,Aruba%20Mobility%20Gateways;softwareGroups=ArubaOS;softwareMajorVersions=8.10

11.2.1 Related Hewlett Packard Enterprise Documents

The following Hewlett Packard Enterprise documents can be referenced to ensure that ArubaOS and the Hewlett Packard Enterprise hardware-based equipment or Hewlett Packard Enterprise virtual appliances that run ArubaOS are installed and operated correctly in the Approved mode:

- *Aruba Access Points Installation Guides*
- *ArubaOS 8.10.0.x AP Software Quick Start Guide*
- *ArubaOS 8.10.0.0 Virtual Appliance Installation Guide*
- *ArubaOS 8.10.0.0 User Guide*
- *ArubaOS 8.10.0.x CLI Reference Guide*
- *ArubaOS 8.10.0.0 API Guide*
- *ArubaOS 8.10.0.0 Getting Started Guide*
- *ArubaOS 8.10.0.0 Syslog Reference Guide*

11.2.2 Administrator Guidance

The Crypto Officer must ensure that the module is kept in the Approved mode of operation. To keep the module in the Approved mode, abide by section 11.1, Start-up Procedures, section 2.9, Non-Approved Algorithms Not Allowed in the Approved Mode of Operation, and section 4.3.2, Non-Approved Services.

11.2.3 Non-Administrator Guidance

None

11.2.4 Maintenance Requirements

Not Applicable (N/A)

11.3 End of Life

To determine if an Hewlett Packard Enterprise product is considered end of life, refer to the Hewlett Packard Enterprise end-of life information at https://networkingsupport.hpe.com/end-of-life. If an Hewlett Packard Enterprise product is deemed end-of-life, the CO should work with their Hewlett Packard Enterprise representative to determine the appropriate Hewlett Packard Enterprise product upgrade path to use a newer Approved version.

For secure sanitization and zeroization of SSP values, the CO should follow the guidance in the Zeroization service entry above in Table 13, Approved Services to wipe out host flash and reboot the module. Since the module is a component of ArubaOS, if the module is deprecated, the module will be upgraded to a newer Approved validated version by loading and booting a newer validated version of ArubaOS with the help of an Hewlett Packard Enterprise-Certified Mobility Professional (ACMP).

12 Mitigation of Other Attacks

The module has not been purposely designed, built and publicly documented to mitigate one or more specific attacks. The Mitigation of Other Attacks requirements are not applicable, per FIPS 140-3 IG 12.A.