FIPS 140-2 Non-Proprietary Security Policy

Aruba Networks Common Cryptographic Module Version 1.0

Document Version 1.9

February 22, 2017

Prepared For:

Aruba, a Hewlett Packard Enterprise Company

3333 Scott Blvd.

Santa Clara, CA 95054

www.arubanetworks.com

Prepared By:

Apex Assurance Group, LLC

530 Lytton Avenue, Ste. 200

Palo Alto, CA 94301

www.apexassurance.com

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the Aruba Networks Common Cryptographic Module Version 1.0.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 About FIPS 140

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Communications Security Establishment of Canada (CSEC) Cryptographic Module Validation Program (CMVP) runs the FIPS 140 program. The CMVP accredits independent testing labs to perform FIPS 140 testing; the CMVP also validates test reports for modules meeting FIPS 140 validation. *Validated* is the term given to a product that is documented and tested against the FIPS 140 criteria.

More information is available on the CMVP website at http://csrc.nist.gov/groups/STM/cmvp/index.html.

1.2 About this Document

This non-proprietary Cryptographic Module Security Policy for the Aruba Networks Common Cryptographic Module Version 1.0 provides an overview of the product and a high-level description of how it meets the security requirements of FIPS 140-2. This document contains details on the module's cryptographic keys and critical security parameters. This Security Policy concludes with instructions and guidance on running the module in a FIPS 140-2 mode of operation.

The Aruba Networks Common Cryptographic Module Version 1.0 may also be referred to as the "module" in this document.

1.3 External Resources

The Aruba Networks website (http://www.arubanetworks.com) contains information on Aruba Networks products. The Cryptographic Module Validation Program website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains links to the FIPS 140-2 certificate and Aruba Networks contact information.

1.4 Notices

This document may be freely reproduced and distributed in its entirety without modification.

1.5 Acronyms

The following table defines acronyms found in this document:

Acronym	Term
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSE	Communications Security Establishment, Canada
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
GUI	Graphical User Interface
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
MD	Message Digest
NIST	National Institute of Standards and Technology
OS	Operating System
PKCS	Public-Key Cryptography Standards
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
Triple-DES	Triple Data Encryption Algorithm
TLS	Transport Layer Security
USB	Universal Serial Bus

Table 1 – Acronyms and Terms

2 Aruba Networks Common Cryptographic Module Version 1.0

2.1 Cryptographic Module Specification

The module, the Aruba Networks Common Cryptographic Module Version 1.0, is a software shared library that provides cryptographic services required by Aruba Networks software applications. The Module's logical cryptographic boundary is the shared dynamic library files (ancrypto.dll, libancrypto.so, and libancrypto.a) and their integrity check HMAC files.

The module is a multi-chip standalone embodiment installed on a General Purpose Computer.

2.1.1 Validation Level Detail

The following table lists the level of validation for each area in FIPS 140-2:

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 2 – Validation Level by DTR Section

2.1.2 Approved Cryptographic Algorithms

The module's cryptographic algorithm implementations have received the following certificate numbers from the Cryptographic Algorithm Validation Program:

Algorithm	Modes/Key Sizes	Security Strength (bits)	CAVP Certificate
AES	CBC, CTR, and GCM[1] modes; E/D; 128, 192 and 256	128, 192, 256	#2744, 2746
Triple-DES	3-key; TCBC mode; E/D	112	#1652, 1653
HMAC	HMAC-SHA-1, -SHA-256, -SHA-384, -SHA-512	112, 128, 192, 256	#1721, 1722
SHA	SHA-1, SHA-256, SHA-384, SHA-512	80 to 256	#2316, 2317

[1] The module's AES GCM implementation meets IG A.5. The IV is generated internally and randomly using the Approved DRBG.

Algorithm	Modes/Key Sizes	Security Strength (bits)	CAVP Certificate
FIPS 186-2 RSA	PKCS #1 1.5 Sig Ver (1024, 2048; SHA-1, 256, 384, 512)	80 or 112	#1483, 1484
FIPS 186-4 RSA	Key Gen (2048); PKCS #1 1.5 Sig Gen (2048; SHA-256, 384, 512); Sig Ver (1024, 2048; SHA-1, 256, 384, 512)	112 112 80 or 112	#1483, 1484
RSASP1	RSA Signature Primitive (2048)	112	CVL #265, 266
FIPS 186-4 ECDSA	Key Gen (P-256, P-384); Sig Gen (P-256, P-384; SHA-256, 384, 512); Sig Ver (P-256, P-384; SHA-1, 256, 384, 512)	112 or 128 112 or 128 112 or 128	#499, 500
DRBG	AES-CTR based DRBG (AES-128, AES-192, AES-256) No derivation function	Modified by available entropy	#496, 498

Table 3 – FIPS-Approved Algorithm Certificates

2.1.3 Non-Approved Cryptographic Algorithms

Within the FIPS Approved mode of operation, the module supports the following allowed algorithms:

- Diffie-Hellman using 2048 (for key agreement; provides 112 bits of encryption strength – compliant with IG D.8 with respect to IG D.11 for IKE)
- RSA Key Wrapping using 2048 (provides 112 bits of encryption strength)
- ECDH using P-224 or P-384 (for key agreement; provides 128 or 192 bits of encryption strength)

In addition to the above algorithms, the following algorithms are available in the non-FIPS Approved mode of operation:

- DES, Blowfish, ARC2, ARC4, MD2, MD4, MD5, HMAC-MD5, AES EAX, AES XCBC, AES MAC (non-compliant)
- RSA PKCS #1 v2.1 RSAES-OAEP encryption/decryption

During operation, the module can switch service by service between an Approved mode of operation and a non-Approved mode of operation. The module will transition to the non-Approved mode of operation when one of the above non-Approved security functions is utilized in lieu of an Approved one. The module can transition back to the Approved mode of operation by utilizing an Approved security function.

The conditions for using the module in an Approved mode of operation are:

1. The module is a cryptographic library and it is intended to be used with a calling application. The calling application is responsible for the usage of the primitives in the correct sequence.

2. The module relies on an entropy source external to the module boundary. The module contains an Approved DRBG which generates random strings whose strengths are modified by available entropy. The DRBG implementation can only be used if full entropy is provided by the calling application.

3. The keys used by the module for cryptographic purposes are determined by the calling application. The calling application is required to provide keys in accordance with FIPS 140-2 requirements.

2.2 Module Interfaces

The figure below shows the module's physical and logical block diagram:

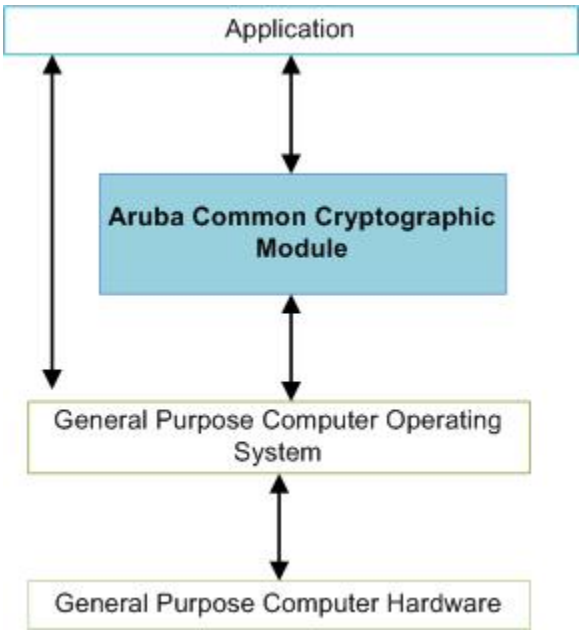

Figure 1 – Module Boundary and Interfaces Diagram

All operations of the module occur via calls from Aruba applications and their respective internal daemons/processes. As such there are no untrusted services calling the services of the module, as APIs are not exposed to any processes except the Aruba components. This module is linked to the calling Aruba application and no interface is provided to the user, so only functions within the module can call the cryptographic functions.

The physical boundary of the module is the surface of the system case that the module resides in.

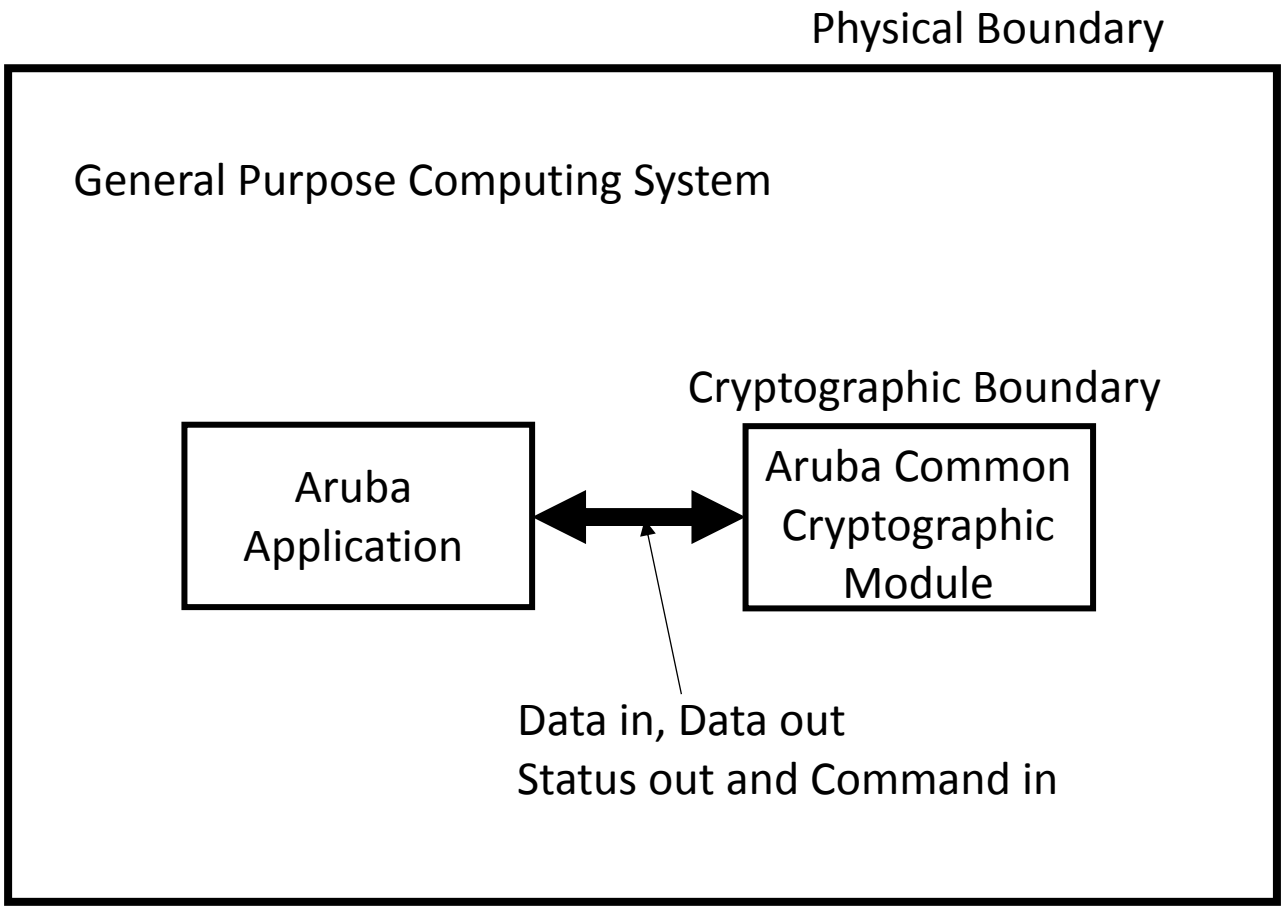

Figure 2 – Logical Cryptographic Boundary/Block Diagram

The logical boundary of the module consists of the extents of the library files which make up the module. The physical boundary is the surface of the case and the modifiable operational environment runs completely within the physical boundary.

The interface ports (shown below in Table 4) include the keyboard, CDROM drive, floppy disk, mouse, network port, parallel port, USB ports, monitor port and power connection. The module's interfaces are logical and are provided through the Application Programming Interface (API) that a calling program can access. The logical interfaces expose services that applications directly call, and the API provides functions that may be called by a referencing application (see Section 2.3 – Roles, Services, and Authentication for the list of available functions). The module distinguishes between the logical interfaces by logically separating the information according to the defined API.

The API provided by the module is mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output. Each of the FIPS 140-2 logical interfaces relates to the module's callable interface, as follows:

FIPS 140-2 Interface	Logical Interface	Module Physical Interface
Data Input	Input parameters of API function calls	Network Interface
Data Output	Output parameters of API function calls	Network Interface
Control Input	API function calls	Command input interfaces
Status Output	For FIPS mode, function calls returning status information and return codes provided by API function calls.	Display Controller
Power	None	Power Supply

Table 4 – Logical Interface / Physical Interface Mapping

As shown in Figure 1 – Module Boundary and Interfaces Diagram, the output data path is provided by the data interfaces and is logically disconnected from processes performing key generation or zeroization. No key information will be output through the data output interface when the module zeroizes keys. No key information is output during key generation.

2.3 Roles, Services, and Authentication

The module supports a Crypto Officer and a User role. The module does not support a Maintenance role. The supported role definitions are as follows:

Role	Services
User	• Self-tests • Show Status
Crypto Officer	• DH Key Generation • DH Key Exchange • ECDH Key Generation • ECDH Key Exchange • RSA Key Generation • RSA Signature Generation • RSA Signature Verification • RSA Key Wrapping Encryption/Decryption • ECDSA Key Generation • ECDSA Signature Generation • ECDSA Signature Verification • AES Encryption/Decryption • Triple-DES Encryption/Decryption • SHA-1 • SHA-256 • SHA-384/512 • HMAC-SHA1 Message Authentication Code • HMAC-SHA256 Message Authentication Code • HMAC-SHA384/512 Message Authentication Code • AES-CTR DRBG Random Number Generation • Key Destruction
No Role Required	• Self-tests invoked by reloading the library into executable memory

Table 5 – Role Descriptions/Approved Services

The User and Crypto-Officer roles are implicitly assumed by the entity accessing services implemented by the Module.

Role	Services
User	• Read Version
Crypto Officer	• DES Encryption • DES Decryption • AES Message Authentication Code • Blowfish Encryption • Blowfish Decryption • ARC2, ARC4 Encryption • ARC2, ARC4 Decryption • MD2 Hash • MD4 Hash • MD5 Hash • HMAC-MD5 Message Authentication Code • AES EAX Encryption • AES EAX Decryption • AES XCBC Encryption • AES XCBC Decryption • RSA PKCS #1 v2.1 RSAES-OAEP Encryption • RSA PKCS #1 v2.1 RSAES-OAEP Decryption

Table 6 – Role Descriptions/Non-Approved services

2.3.1 Operator Services and Descriptions

The module supports services that are available to the User and Crypto Officer (administrator) roles. All of the services are described in detail in the module's user documentation. The following table shows the Approved/allowed services available, the access to cryptographic keys and CSPs, and the status resulting from services:

Role	Approved/Allowed Service	Cryptographic Keys and CSP Access
Crypto Officer	DH Key Generation	• Use DH Public and Private Components • Generate DH Key pair
Crypto Officer	DH Key Exchange	• Use DH Private Component • Generate DH shared secret
Crypto Officer	ECDH Key Generation	• Use ECDH Public and Private Components • Generate ECDH Key pair
Crypto Officer	ECDH Key Exchange	• Use ECDH Private Component • Generate ECDH shared secret
Crypto Officer	RSA Key Generation	• Generate RSA Public/Private Key pair
Crypto Officer	RSA Signature Generation	• Use RSA Private Key • Generate RSA Signature
Crypto Officer	RSA Signature Verification	• Use RSA Public Key • Verify RSA Signature

Role	Approved/Allowed Service	Cryptographic Keys and CSP Access
Crypto Officer	RSA Key Wrapping Encryption	• Use RSA Public Key • Performs Key Wrapping Encryption
Crypto Officer	RSA Key Wrapping Decryption	• Use RSA Private Key • Performs Key Wrapping Decryption
Crypto Officer	ECDSA Key Generation	• Generate ECDSA Key Pair for Signature Generation/Verification
Crypto Officer	ECDSA Signature Generation	• Use ECDSA Private Key • Generate ECDSA Signature
Crypto Officer	ECDSA Signature Verification	• Use ECDSA Public Key • Verify ECDSA Signature
Crypto Officer	AES Encryption	• Use AES Key
Crypto Officer	AES Decryption	• Use AES Key
Crypto Officer	Triple-DES Encryption	• Use Triple-DES Key
Crypto Officer	Triple-DES Decryption	• Use Triple-DES Key
Crypto Officer	SHA-1	• Generate SHA-1 Output; no CSP access
Crypto Officer	SHA-256	• Generate SHA-256 Output; no CSP access
Crypto Officer	SHA-384/512	• Generate SHA-384/512 Output; no CSP access
Crypto Officer	HMAC-SHA-1 Message Authentication Code	• Use HMAC-SHA-1 Key • Generate HMAC-SHA-1 Output
Crypto Officer	HMAC-SHA-256 Message Authentication Code	• Use HMAC-SHA-256 Key • Generate HMAC-SHA-256 Output
Crypto Officer	HMAC-SHA-384/512 Message Authentication Code	• Use HMAC-SHA-384/512 Key • Generate HMAC-SHA-384/512 Output
Crypto Officer	AES-CTR DRBG Random Number Generation	• Use V and “Key” values to generate random number • Destroy V and “Key” values after use
Crypto Officer	Key Destruction	• Destroy All CSPs
User	Show Status	N/A
User	Self-Tests	N/A

Table 7 – Approved/Allowed Module Services and Descriptions

Note: Table 7 above does not include the non-Approved services supported by the module.

2.3.2 Operator Authentication

As required by FIPS 140-2, there are two roles (a Crypto Officer role and User role) in the module that operators may assume. As allowed by Level 1, the module does not support authentication to access services.

2.4 Physical Security

This section of requirements does not apply to this module. The module is a software-only module and does not implement any physical security mechanisms.

2.5 Operational Environment

The module operates on a general purpose computer (GPC) running Microsoft Windows 7, Linux/Android, or iOS as a general purpose operating system (GPOS). For FIPS purposes, the module is running on this operating system in single user mode and does not require any additional configuration to meet the FIPS requirements.

The module was tested on the following platforms:

- Android 4 (ARMv7)
- Windows 7 Enterprise User Mode w/ SP1 (32-bit x86)
- Windows 7 Enterprise User Mode w/ SP1 (64-bit x86_64)
- Red Hat Enterprise Linux 6 with Linux 2.6 kernel (32-bit x86)
- iOS 9

Compliance is maintained for other versions of the respective operating systems family where the binary is unchanged.

The GPC(s) used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B. FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the GPOS running in single user mode, assuming that the requirements outlined in NIST IG G.5 are met.

2.6 Cryptographic Key Management

The table below provides a complete list of Critical Security Parameters used within the module:

R = Read W = Write D = Delete

CSP	CSP Description/ Usage	Generation	Storage	Entry/Output	R/W	Destruction
DH Private Components	Used to derive the secret session key during DH key agreement protocol Group 14 (384 bits)	Internally using the AES-CTR DRBG	Plaintext in volatile RAM	N/A	R/W	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
DH Shared Secret	Diffie-Hellman secret key (2048 bits)	Established during Diffie-Hellman Exchange	Plaintext in volatile RAM	N/A	R/W	Zeroized after the session is closed.

CSP	CSP Description/ Usage	Generation	Storage	Entry/Output	R/W	Destruction
ECDH Private Components	Used to derive the secret session key during ECDH key agreement protocol (Group 19 (P-256) and Group 20 (P-384))	Internally using the AES-CTR DRBG	Plaintext in volatile RAM	N/A	R/W	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
ECDH Shared Secret	Elliptic Curve Diffie-Hellman secret key (P-256 and P-384)	Established during EC Diffie-Hellman Exchange	Plaintext in volatile RAM	N/A	R/W	Zeroized after the session is closed.
DRBG V and "Key" values	DRBGs for key generation DRBG V: SP800-90a DRBG (384 bits) DRBG "Key": SP800-90a (256 bits)	Externally	Plaintext in volatile RAM	Entry: Plaintext, Electronic, from host OS. Output: N/A	W	Automatically after use
RSA Private Key	Used to create RSA digital signatures Key size: 2048	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext	R/W	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
RSA Key Wrapping Private Key	Used for RSA Key Wrapping decryption operation Key size: 2048	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext		An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.

CSP	CSP Description/ Usage	Generation	Storage	Entry/Output	R/W	Destruction
ECDSA Private Key	Used to create DSA digital signatures Key size: 256, 384	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext		An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
Triple-DES Key	Used during Triple-DES encryption and decryption Key size: 192 bit	Externally	Plaintext in volatile RAM	Entry: Plaintext Output: N/A		An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
AES Keys	Used during AES encryption and decryption Key length: 128, 192, 256	Externally	Plaintext in volatile RAM	Entry: Plaintext Output: N/A		An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
HMAC Keys	Used during HMAC SHA-1, 256, 384, 512 operations Key size: 192, 256, 384, 512	Externally	Plaintext in volatile RAM	Entry: Plaintext Output: N/A		An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
DH Public Component	Used to derive the secret session key during DH key agreement protocol DH Groups: Group 14 (2048 bits)	Internally using the AES-CTR DRBG	Plaintext in volatile RAM	Entry: Receive Client Public Component during DH exchange. Output: Transmit Host Public Component during DH exchange		N/A

CSP	CSP Description/ Usage	Generation	Storage	Entry/Output	R/W	Destruction
ECDH Public Component	Used to derive the secret session key during ECDH key agreement protocol Group 19 (P-256), Group 20 (P-384)	Internally using the AES-CTR DRBG	Plaintext in volatile RAM	Entry: Receive Client Public Component during DH exchange. Output: Transmit Host Public Component during DH exchange		N/A
RSA Public Keys	Used to verify RSA Signatures Key size: 2048	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext		N/A
RSA Key Wrapping Public Keys	Used for RSA Key Wrapping encryption operation Key size: 2048	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext		N/A
ECDSA Public Keys	Used to verify ECDSA signatures Key size: 256, 384	Internally using the AES-CTR DRBG or generated externally	Plaintext in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext		N/A

Table 8 – Module Keys/CSPs

The application that uses the module is responsible for appropriate destruction and zeroization of the key material. The library provides functions for key allocation and destruction which overwrites the memory that is occupied by the key information with zeros before it is deallocated.

The min entropy measurement from the Linux kernel source is **0.98683625** bits of entropy per binary digit as measured using the NIST Python tool on a 1 Mbyte sample.

In operation, the approved DRBG used in the module is seeded with 64 bytes (512 bits) of entropy from the Linux entropy pool, yielding 512 * **0.986383625** = 505.26016 bits of entropy per seeding operation, which is larger than the 256 bits entropy required.

2.7 Self-Tests

FIPS 140-2 requires that the module perform self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition some functions require continuous verification of function, such as the random number generator. All of these tests are listed and described in this section. In the event of a self-test error, the module will log the error and will halt. The module must be initialized into memory to resume function.

The following sections discuss the module's self-tests in more detail.

2.7.1 Power-On Self-Tests

Power-on self-tests are executed automatically when the module is loaded into memory. All encrypt & decrypt tests are performed separately.

TYPE	DETAIL
Cryptographic Algorithm Tests	• AES-CBC, GCM, CCM encrypt & decrypt KATs • Triple-DES encrypt & decrypt KATs • HMAC-SHA-1, -256, -384, -512 KATs • SHA-1, -256, -384, -512 KATs • RSA Sign and Verify KATs • RSA encrypt & decrypt KAT • ECDSA Pairwise Consistency Test • DH Pairwise Consistency Test • ECDH Pairwise Consistency Test • AES-CTR DRBG KAT and Health Tests (generate function is tested in every instance that the module is loaded into memory)
Module integrity	• HMAC-SHA-1

Table 9 – Power-On Self-Tests

Input, output, and cryptographic functions cannot be performed while the Module is in a self-test or error state because the module is single-threaded and will not return to the calling application until the power-up self tests are complete. If the power-up self-tests fail, subsequent calls to the module will also fail - thus no further cryptographic operations are possible.

2.7.2 Conditional Self-Tests

The module implements the following conditional self-tests upon key generation, or random number generation (respectively):

TYPE	DETAIL
Pair-wise Consistency Tests	• RSA • ECDSA
DRBG continuous Test	• AES-CTR DRBG (as defined in sec 4.92 of FIPS 140-2)

Table 10 – Conditional Self-Tests

2.8 Mitigation of Other Attacks

The Module does not contain additional security mechanisms beyond the requirements for FIPS 140-2 Level 1 cryptographic modules.

3 Guidance and Secure Operation

This section describes how to configure and initialize the module for FIPS-Approved mode of operation. When configured and initialized per this Security Policy, the module will only operate in the FIPS Approved mode of operation.

3.1 Crypto Officer Guidance

3.1.1 Software Installation

The module is not available for direct download. The module is to be installed on a single-user mode operating system specified in Section 2.5 or one where portability is maintained.

3.1.2 Key Destruction Service

There is a context structure associated with every cryptographic algorithm available in this module. Context structures hold sensitive information such as cryptographic keys. These context structures must be destroyed via respective API calls when the application software no longer needs to use a specific algorithm any more. This API call will zeroize all sensitive information including cryptographic keys before freeing the dynamically allocated memory.

3.2 Additional Rules of Operation

The module complies with the following security rules:

- The cryptographic module shall provide two distinct roles - User role and the Cryptographic Officer role.
- The cryptographic module does not provide any operator authentication.
- The cryptographic module shall encrypt/decrypt message traffic using the Triple-DES or AES algorithms.
- The cryptographic module shall perform all self-tests described in Section 7.
- The cryptographic module is available to perform services only after successfully completing the power-up self-tests.
- At any time, the operator shall be capable of commanding the module to perform the power-up self-tests by reloading the cryptographic module into memory.
- Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
- Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
- The module shall not support concurrent operators.
- DES, Blowfish, ARC2, ARC4, MD2, MD4, MD5, HMAC-MD5, AES EAX, AES XCBC, and RSA PKCS #1 v2.1 RSAES-OAEP encryption/decryption are not allowed for use in the FIPS Approved mode of operation. It is the responsibility of the consuming application to zeroize all keys and CSPs prior to and after utilizing these non-Approved algorithms. CSPs shall not be shared between the Approved and non-Approved modes of operation.

3.3 User Guidance

3.3.1 General Guidance

The module is not distributed as a standalone library and is only used in conjunction with Aruba Networks solutions. As such, there is no direct User Guidance.