FIPS 140-2 Non-Proprietary Security Policy

Poly Crypto Module for MobileOS

Software Version 1.0

Document Version 1.0

July 16, 2021

Prepared For:

Plantronics, Inc.
345 Encinal Street
Santa Cruz, CA 95060
www.poly.com

Prepared By:

SafeLogic Inc.
530 Lytton Ave, Suite 200
Palo Alto, CA 94301
www.safelogic.com

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for Crypto Module for MobileOS.

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 Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140 program. The NVLAP accredits independent testing labs to perform FIPS 140 testing; the CMVP validates modules meeting FIPS 140 validation. *Validated* is the term given to a module 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 Crypto Module for MobileOS from Plantronics, Inc. ("Poly") 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.

Poly Crypto Module for MobileOS may also be referred to as the "module" in this document.

1.3 External Resources

The Poly website (www.poly.com) contains information on Poly services and products. The Cryptographic Module Validation Program website contains links to the FIPS 140-2 certificate and Poly 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
BT	BlueTooth
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CCCS	Canadian Centre for Cyber Security
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Number Generator
DSA	Digital Signature Algorithm
EC	Elliptic Curve
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GPD	General Purpose Device
GUI	Graphical User Interface
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
MD	Message Digest
NVLAP	National Voluntary Laboratory Accreditation Program
NIST	National Institute of Standards and Technology
OS	Operating System
PKCS	Public-Key Cryptography Standards
PRNG	Pseudo Random Number Generator
PSS	Probabilistic Signature Scheme
RF	Radio Frequency
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 Poly Crypto Module for MobileOS

2.1 Cryptographic Module Specification

Poly Unified Communications Cryptographic Module for MobileOS is a standards-based "Drop-in Compliance" cryptographic engine for servers and appliances. The module delivers core cryptographic functions to mobile platforms and features robust algorithm support, including Suite B algorithms. Poly Unified Communications Cryptographic Module for MobileOS offloads functions for secure key management, data integrity, data at rest encryption, and secure communications to a trusted implementation.

The module's logical cryptographic boundary is the shared library files and their integrity check HMAC files. The module is a multi-chip standalone embodiment installed on a General Purpose Device.

All operations of the module occur via calls from host applications and their respective internal daemons/processes. As such there are no untrusted services calling the services of the module.

The module supports two modes of operation: Approved and non-Approved. The module will be in the FIPS-approved mode when all power up self-tests have completed successfully, and only Approved algorithms are invoked. See *Approved Cryptographic Algorithms* section below for a list of the supported Approved algorithms. The non-Approved mode is entered when a non-Approved algorithm is invoked. See *Non-Approved Algorithms* for a list of non-Approved algorithms.

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	3
Mitigation of Other Attacks	N/A

Table 2 – Validation Level by FIPS 140-2 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	CAVP Certificate for iOS	CAVP Certificate for Android
AES **ECB** (e/d; 128 , 192 , 256) **CBC** (e/d; 128 , 192 , 256) **CFB1** (e/d; 128 , 192 , 256) **CFB8** (e/d; 128 , 192 , 256) **OFB** (e/d; 128 , 192 , 256) **CTR** (ext only; 128 , 192 , 256) **CCM** (KS: 128 , 192 , 256) **CMAC** (Generation/Verification) (KS: 128, 192, 256) **GCM** (KS: AES_128(e/d), AES_192(e/d), AES_256(e/d))	2126	2125
HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC- SHA-384, HMAC-SHA-512	1297	1296
DSA FIPS 186-4 **PQG Gen**: 2048 & 3072 (using SHA-2) **PQG Ver**: 1024, 2048 & 3072 (using SHA-1 and SHA-2) **Key Pair**: 2048-bit & 3072-bit **Sig Gen**: 2048-bit & 3072-bit (using SHA-2) **Sig Ver**: 1024-bit, 2048-bit & 3072-bit (using SHA-1 and SHA-2)	667	666
ECDSA FIPS 186-4 **Key Pair Generation**: Curves (P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 & B-571) **PKV**: Curves All P, K & B **Sig Gen**: (P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 & B-571) (using SHA-2) **Sig Ver**: Curves P-192, P224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 & B-571 (using SHA-1 and SHA-2)	320	319

Algorithm	CAVP Certificate for iOS	CAVP Certificate for Android
RSA (X9.31, PKCS #1.5, PSS) FIPS 186-2 **ANSIX9.31** Sig Gen: 4096 bit (using SHA-2) Sig Ver: 1024-bit, 1536-bit, 2048-bit, 3072-bit, 4096-bit (any SHA size) **PKCS1 V1 5** Sig Gen: 4096-bit (using SHA-2) Sig Ver: 1024-bit, 1536-bit, 2048-bit, 3072-bit, 4096-bit (any SHA size) **PSS** Sig Gen: 4096-bit (using SHA-2) Sig Ver: 1024-bit, 1536-bit, 2048-bit, 3072-bit, 4096-bit (any SHA size) FIPS 186-4 **ANSIX9.31** Sig Gen: 2048-bit & 3072-bit (using SHA-2) Sig Ver: 1024-bit, 2048-bit, & 3072-bit (any SHA size) **PKCS1 V1 5** Sig Gen: 2048-bit & 3072-bit (using SHA-2) Sig Ver: 1024-bit, 2048-bit, & 3072-bit (any SHA size) **PSS** Sig Gen: 2048-bit & 3072-bit (using SHA-2) Sig Ver: 1024-bit, 2048-bit, & 3072-bit (any SHA size)	1095	1094
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1850	1849
Triple-DES **TECB**(KO 1 e/d, KO 2 d only) **TCBC**(KO 1 e/d, KO 2 d only) **TCFB1**(KO 1 e/d, KO 2 d only) **TCFB8**(KO 1 e/d, KO 2 d only) **TCFB64**(KO 1 e/d, KO 2 d only) **TOFB**(KO 1 e/d, KO 2 d only) **CMAC**(KS: 3-Key; Generation/Verification; Block Size(s): Full / Partial)	1352	1351
SP 800-90A Rev.1 DRBG (Hash_DRBG, HMAC_DRBG, CTR_DRBG)	234	233
CKG	Vendor Affirmed	

Table 3 – FIPS-Approved Algorithm Certificates

2.1.3 Non-Approved Mode of Operation

The following algorithms shall not be used:

- AES XTS ((KS: XTS_128((e/d) (f/p)) KS: XTS_256((e/d) (f/p))
- EC Diffie-Hellman
- RSA (key wrapping; key establishment methodology provides up to 256 bits of encryption strength)
- GMAC

The following algorithms are disallowed as of January 1, 2016 per the NIST SP 800-131A algorithm transitions:

- Random Number Generator Based on ANSI X9.31 Appendix A.2.4
- Two-Key Triple DES Encryption
- Dual EC DRBG

The following algorithms are disallowed as of January 1, 2014 per the NIST SP 800-131A algorithm transitions:

- FIPS 186-4 DSA
  PQG Gen 1024-bit (any SHA size), 2048-bit & 3072-bit using SHA-1
  Key Gen 1024-bit (any SHA size), 2048-bit & 3072-bit using SHA-1
  Sig Gen 1024-bit (any SHA size), 2048-bit & 3072-bit using SHA-1
- FIPS 186-2 DSA
  PQG Gen 1024-bit (any SHA size)
  PQG Ver 1024-bit
  Key Gen 1024-bit
  Sig Gen 1024-bit (any SHA size), 2048-bit & 3072-bit using SHA-1
- FIPS 186-2 RSA
  ANSIX9.31
  Key Gen 1024 & 1536
  ANSIX9.31
  Sig Gen 1024 & 1536 (any SHA size); 2048, 3072 & 4096 using SHA-1
  PKCSI V1 5
  Sig Gen 1024 & 1536 (any SHA size) 2048, 3072 & 4096 using SHA-1
  PSS
  Sig Gen 1024 & 1536 (any SHA size) 2048, 3072 & 4096 using SHA-1
- FIPS 186-4 RSA
  ANSIX9.31
  Sig Gen 1024 using SHA-1
  PKCSI V1 5
  Sig Gen 1024 using SHA-1

PSS
Sig Gen 1024 using SHA-1

- FIPS 186-2 ECDSA — **Key Pair Generation: Curves** P-192, K-163 & B-163
  PKV: Curves All P, K & B
  Sig Gen Curves All P, K & B
  Sig Ver Curves All P, K & B

- FIPS 186-4 ECDSA — **Key Pair Generation: Curves** P-192, K-163 & B-163
  Sig Gen Curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 & B-571 (using SHA-1)
  P-192, K-163 & B-163 (any SHA size)
- CVL (ECC CDH KAS)

The following algorithms are disallowed as of September 1, 2020 per the FIPS 186-2 transitions:

- FIPS 186-2 RSA (X9.31, PKCS #1.5, PSS)
  - **ANSIX9.31**
    - Key Gen: 2048-bit, 3072-bit & 4096-bit
    - Sig Gen: 2048-bit, 3072-bit (any SHA size)
    - Sig Gen: 4096-bit using SHA-1
  - **PKCS1 V1 5**
    - Sig Gen: 2048-bit, 3072-bit (any SHA size)
    - Sig Gen: 4096-bit using SHA-1
  - **PSS**
    - Sig Gen: 2048-bit, 3072-bit (any SHA size)
    - Sig Gen: 4096-bit using SHA-1

2.2 Module Interfaces

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

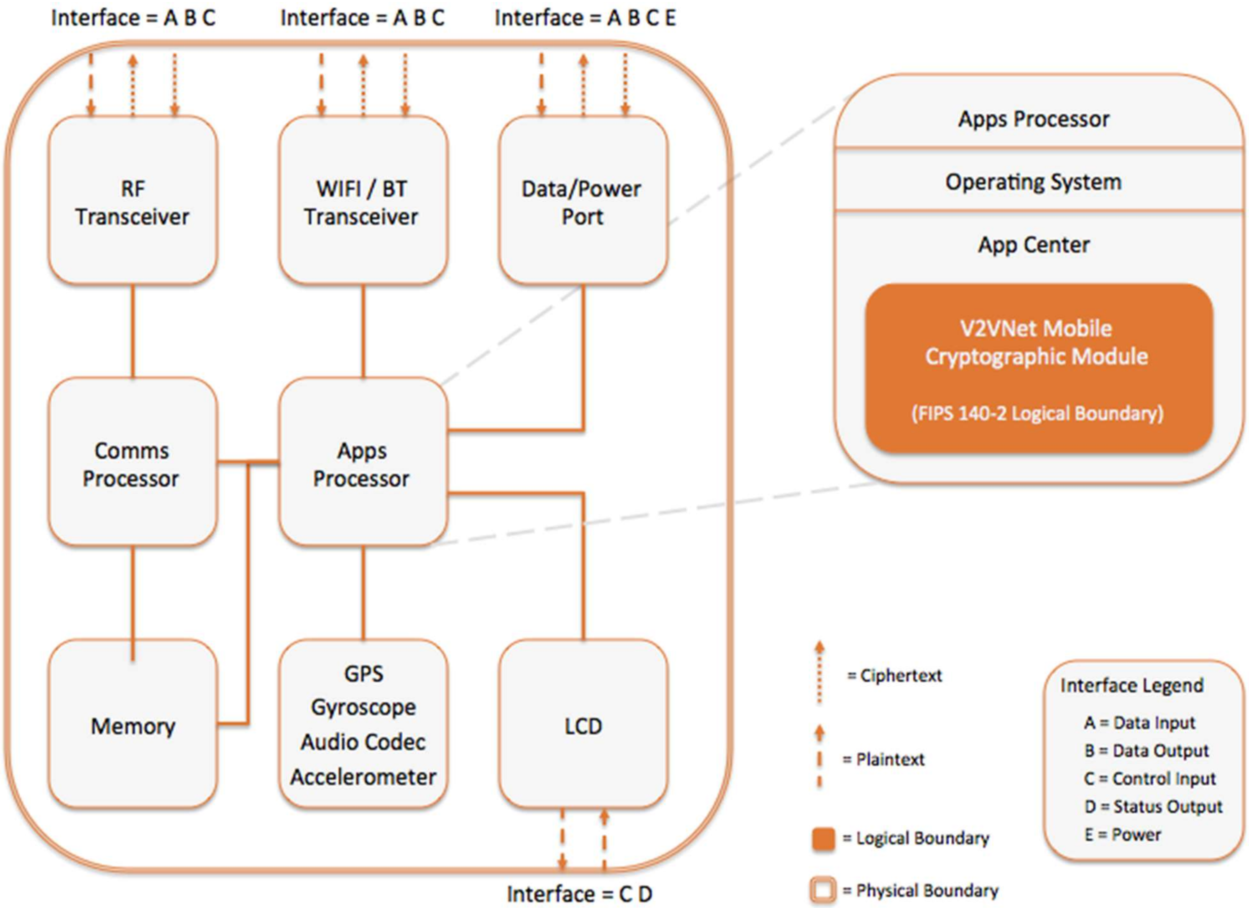

Figure 1 – Module Boundary and Interfaces Diagram

The interfaces (ports) for the physical boundary include the Data/power port, WIFI / BT Transceiver, RF Transceiver, and LCD. When operational, the module does not transmit any information across these physical ports because it is a software cryptographic module. Therefore, the module's interfaces are purely logical and are provided through the Application Programming Interface (API) that a calling daemon can operate. 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 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	Data/power port WIFI / BT Transceiver RF Transceiver
Data Output	Output parameters of API function calls	Data/power port WIFI / BT Transceiver RF Transceiver
Control Input	API function calls	Data/power port WIFI / BT Transceiver RF Transceiver LCD
Status Output	For FIPS mode, function calls returning status information and return codes provided by API function calls.	LCD
Power	None	Data/power port

Table 4 – Logical Interface / Physical Interface Mapping

As shown in Figure 1 – Module Boundary and Interfaces Diagram and Table 5 – Module Services, Roles, and Descriptions, 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.

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

2.3.1 Operator Services and Descriptions

The module supports services that are available to users in the various roles. All of the services are described in detail in the module's user documentation. The following table shows the services available to the various roles and the access to cryptographic keys and CSPs resulting from services:

Service	Roles	CSP / Algorithm	Permission
Module initialization	Crypto Officer	None	CO: execute
Symmetric encryption/decryption	User	AES Key, Triple-DES Key	User: read/write/execute
Digital signature generation	User	RSA Private Key, DSA Private Key, ECDSA Private Key	User: read/write/execute

Service	Roles	CSP / Algorithm	Permission
Digital Signature verification	User	RSA Public Key, DSA Public Key, ECDSA Public Key	User: read/write/execute
Symmetric key generation	User	AES Key, Triple-DES Key	User: read/write/execute
Asymmetric key generation	User	DSA Private Key, ECDSA Private Key	User: read/write/execute
Keyed Hash (HMAC)	User	HMAC Key HMAC SHA-1, HMAC SHA- 224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512	User: read/write/execute
Message digest (SHS)	User	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	User: read/write/execute
Random number generation	User	DRBG Internal State, DRBG Entropy	User: read/write/execute
Show status	Crypto Officer User	None	User and CO: execute
Self test	User	None	User: read/execute
Zeroize	Crypto Officer User	All CSPs	CO: read/write/execute

Table 5 – Module Services, Roles, and Descriptions

The operator is required to review the sections Approved Cryptographic Algorithms, Non-Approved Cryptographic Algorithms, and Guidance and Secure Operation to ensure only approved algorithms are used.

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. As such, there are no applicable authentication policies. Access control policies are implicitly defined by the services available to the roles as specified in Table 5 – Module Services, Roles, and Descriptions.

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 device (GPD) running 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.

Platform	Operating System	CPU(s)
iPad 3	iOS 5.1, 6, 7	ARM A5X
Galaxy Nexus	Android 4.0	ARM Cortex-A9

Table 6 – Tested Environments

Compliance is maintained for other versions of the respective operating system family where the binary is unchanged. No claim can be made as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate.

The GPD(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:

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
AES Key (128, 192, 256 bits) Encrypt/Decrypt operations Used to generate and verify MACs with AES as part of the CMAC algorithm.	RAM	Plaintext	API call parameter	None	power cycle cleanse()	CO: RWD U: RWD

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
Triple-DES Key (168 bits) Used for Encrypt/Decrypt operations. Used for generating and verifying MACs with Triple- DES as part of the CMAC algorithm.	RAM	Plaintext	API call parameter	None	power cycle cleanse()	CO: RWD U: RWD
RSA Public Key (1024, 1536, 2048, 3072, 4096 bits) RSA public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD
RSA Private Key (2048, 3072, 4096 bits) RSA public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD
DSA Public Key (1024, 2048, and 3072 bits) DSA public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
DSA Private Key (2048, and 3072 bits) DSA public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD
HMAC Key (≥ 112 bits) HMAC keys used to generate and verify MACs on data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD
Integrity Key	NVRAM	Plaintext	None	None	None	CO: RWD U: RWD
ECDSA Private Key (**PKG**: Curves (P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 & B-571) **PKV**: Curves All P, K & B) ECDSA keys public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
ECDSA Public Key (**PKG**: Curves (P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 & B-571) **PKV**: Curves All P, K & B) ECDSA keys public/private keys used to sign and verify data.	RAM	Plaintext	API call parameter	API call parameter	power cycle cleanse()	CO: RWD U: RWD
DRBG Internal state (V,C , Key value) V and key are used as part of HMAC and CTR DRBG process. V and C are used as part of HASH DRBG process.	RAM	Plaintext	None	None	power cycle cleanse()	CO: RWD U: RWD
DRBG Entropy Entropy input strings used as part of the DRBG process.	RAM	Plaintext	API call parameter	None	power cycle cleanse()	CO: RWD U: RWD

R = Read W = Write D = Delete

Table 7 – Module Keys/CSPs

Please note that keys can be generated by the module for the services that require those keys, but the keys will always be input via an API call.

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

2.6.1 Random Number Generation

The module uses SP800-90A DRBGs for creation of asymmetric and symmetric keys.

The module accepts input from entropy sources external to the cryptographic boundary for use as seed material for the module's Approved DRBGs. The calling application of the module shall use entropy sources that meet the security strength required for the random bit generation mechanism as shown in NIST Special Publication 800-90A Table 2 (Hash_DRBG, HMAC_DRBG) and Table 3 (CTR_DRBG). At a minimum, the entropy source shall provide at least 128-bits of entropy to the DRBG.

The module performs continual tests on the random numbers it uses to ensure that the seed input to the Approved DRBGs do not have the same value. The module also performs continual tests on the output of the Approved DRBGs to ensure that consecutive random numbers do not repeat.

In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per NIST SP 800-133rev2 (vendor affirmed). The resulting symmetric key or asymmetric seed is an unmodified output from a DRBG.

The AES GCM IV generation is in compliance with the RFC5288 and RFC5289 and shall only be used for the TLS protocol version 1.2 to be compliant with [FIPS140-2_IG] IG A.5, provision 1 ("TLS protocol IV generation"); thus, the module is compliant with [SP800-52].

2.6.2 Key/Critical Security Parameter (CSP) Authorized Access and Use by Role and Service/Function

An authorized application as user (the User role) has access to all key data generated during the operation of the module.

2.6.3 Key/CSP Storage

Public and private keys are provided to the module by the calling process and are destroyed when released by the appropriate API function calls or during power cycle. The module does not perform persistent storage of keys.

2.6.4 Key/CSP Zeroization

The application is responsible for calling the appropriate destruction functions from the API. The destruction functions then overwrite the memory occupied by keys with zeros and deallocates the memory. This occurs during process termination / power cycle. Keys are immediately zeroized upon deallocation, which sufficiently protects the CSPs from compromise.

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. The module verifies the integrity of the runtime executable using a HMAC-SHA1 digest computed at build time. If the fingerprints match, the power-up self-tests are then performed. If the power-up self-test is successful, a flag is set to place the module in FIPS mode.

TYPE	DETAIL
Software Integrity Check	• HMAC-SHA1 on all module components
Known Answer Tests[1]	• AES ECB mode encrypt/decrypt 128-bit key length • AES CCM mode encrypt/decrypt 192-bit key length • AES GCM mode encrypt/decrypt 256-bit key length • AES CMAC CBC mode, encrypt/decrypt with 128, 192, 256-bit key lengths • XTS-AES (legacy test) • EC Diffie-Hellman (legacy test) • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • HMAC-SHA1 • HMAC-SHA224 • HMAC-SHA256 • HMAC-SHA384 • HMAC-SHA512 • RSA sign/verify using 2048 bit key, SHA-256, PKCS#1 • SP 800-90A DRBG (Hash_DRBG, HMAC_DRBG, CTR_DRBG) • Triple-DES ECB mode encrypt/decrypt 3-key • Triple-DES CMAC CBC mode generate/verify 3-key
Pair-wise Consistency Tests	• DSA sign/verify using 2048 bit key, SHA-384 • ECDSA keygen/sign/verify using P-224, K-233 and SHA512 • RSA (legacy test)

[1] Note that all SHA-X KATs are tested as part of the respective HMAC SHA-X KAT. SHA-1 is also tested independently.

Table 8 – 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.

The Module performs power-up self-tests automatically during loading of the module by making use of default entry point (DEP) and no operator intervention is required.

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	• DSA • RSA (legacy test not run in FIPS mode) • ECDSA
Continuous RNG Tests	• Performed on all Approved DRBGs, the non-approved X9.31 RNG, and the non-approved DUAL_EC_DRBG Please note the DRBG is Tested as required by [SP800-90A] Section 11

Table 9 – Conditional Self-Tests

2.7.3 Cryptographic Function

The module verifies the integrity of the runtime executable using a HMAC-SHA1 digest which is computed at build time. If this computed HMAC-SHA1 digest matches the stored, known digest, then the power-up self-test (consisting of the algorithm-specific Pairwise Consistency and Known Answer tests) is performed. If any component of the power-up self-test fails, an internal global error flag is set to prevent subsequent invocation of any cryptographic function calls. Any such power-up self test failure is a hard error that can only be recovered by reinstalling the module[2]. The power-up self-tests may be performed at any time by reloading the module.

No operator intervention is required during the running of the self-tests.

[2] The initialization function could be re-invoked but such re-invocation does not provide a means from recovering from an integrity test or known answer test failure

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

3.1 Crypto Officer Guidance

3.1.1 Software Installation

The module is not available for direct download to the general public. The module and its host application is to be installed on an operating system specified in Section 2.5 or one where portability is maintained.

3.1.2 Additional Rules of Operation

1. The writable memory areas of the module (data and stack segments) are accessible only by the application so that the operating system is in "single user" mode, i.e. only the application has access to that instance of the module.

2. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the module.

3.2 User Guidance

3.2.1 General Guidance

The module is not distributed as a standalone library and is only used in conjunction with the solution.

The end user of the operating system is also responsible for zeroizing CSPs via wipe/secure delete procedures.

If the module power is lost and restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are redistributed.

The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party to encounter this condition shall trigger a handshake to establish a new encryption key in accordance with RFC 5246.

The AES GCM IV generation is in compliance with the RFC5288 and RFC5289 and shall only be used for the TLS protocol version 1.2 to be compliant with [FIPS140-2_IG] IG A.5, provision 1 (“TLS protocol IV generation”); thus, the module is compliant with [SP800-52].

In the event the `nonce_explicit` part of the IV exhausts the maximum number of possible values for a given session key, either party (the client or the server) that encounters this condition shall trigger a handshake to establish a new encryption key.

The same Triple-DES key shall not be used to encrypt more than 2^{16} 64-bit blocks of data in accordance with IG A.13.

At a minimum, the entropy source shall provide at least 128-bits of entropy to the DRBG.

