Plantronics, Inc.

Poly Unified Communications Cryptographic Module

Firmware Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.13

Prepared for:

Plantronics, Inc.
345 Encinal Street
Santa Cruz, CA 95060
United States of America

Phone: +1 831 426 5858
www.plantronics.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Poly Unified Communications Cryptographic Module (firmware version: 1.0) from Plantronics, Inc. (Plantronics). This Security Policy describes how the Poly Unified Communications Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) website (https://csrc.nist.gov/projects/cryptographic-module-validation-program).

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The Poly Unified Communications Cryptographic Module is referred to in this document as Poly UC Crypto Module or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Plantronics website (http://www.plantronics.com/ contains information on the full line of products from Plantronics.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

1.3 Document Organization

The Security Policy document is organized into two primary sections. Section 2 provides an overview of the validated module. This includes a general description of the capabilities and the use of cryptography, as well as a presentation of the validation level achieved in each applicable functional area of the FIPS standard. It also provides high-level descriptions of how the module meets FIPS requirements in each functional area. Section 3 documents the guidance needed for the secure use of the module, including initial setup instructions and management methods and policies.

2. Poly UC Crypto Module

2.1 Overview

Plantronics uses the most advanced audio, video, and content-sharing technologies to create solutions that are powerful, affordable and easy to use. Those solutions address a range of communication activities, from telephony and video conferencing, to telepresence and unified communications (UC).

Plantronics' Poly UC Software the telecommunications industry's broadest and most feature rich software for IP-enabled devices and is designed to optimize business communication while integrating seamlessly into 60+ call control platforms with over 150+ telephony features.

Plantronics' Poly UC Software has an extensive set of features that enable wired and wireless devices from Polycom to integrate seamlessly into a wide variety of unified communications environments. By leveraging an open-standard software architecture, Polycom devices work with the third-party systems of today while keeping options open to easily change direction if needed in the future. Additionally, the Poly UC Software provides a simple-to-use interface for product end-users and IT staff.

Plantronics' Poly UC Software provides the following benefits:

- improves desktop productivity for users at all levels of the organization
- is simple to deploy and is easy to administer, upgrade, and maintain
- leverages existing communication investments, third-party UC software, and productivity applications
- is compatible with multiple industry standard call control platforms, Open SIP, and Skype for Business.
- is compatible with IPv6 when used in Open SIP deployments

With Poly UC Software, end-users can:

- simplify manageability of Polycom devices
- reduce downtime through robust security options
- get out-of-the-box provisioning for enterprises and service providers
- benefit from robust diagnostics for ongoing reporting and tracking
- enjoy simple operation, which reduces training and support costs

The Poly Unified Communications Cryptographic Module provides the cryptographic service support required by Plantronics' Poly Unified Communications Software solutions (including the Poly VideoOS software, versions 3.1 and later). In this document, these software solutions will be referred to collectively as the "calling application". The module is used by the calling application to provide or support symmetric and asymmetric cipher operation, signature generation and verification, hashing, cryptographic key generation, random number generation, message authentication functions, and secure key agreement/key exchange protocols.

The Poly Unified Communications Cryptographic Module is validated at the FIPS 140-2 section levels shown in Table 1.

Table 1 – Security Level per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A[1]
7	Cryptographic Key Management	1
8	EMI/EMC[2]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The Poly Unified Communications Cryptographic Module is a firmware-hybrid module with a multiple-chip standalone embodiment with an overall security level of 1. It is a firmware-hybrid module (based on the BoringCrypto cryptographic library in Google's BoringSSL) providing a C-language application program interface (API) for use by other processes that require cryptographic functionality. The module relies on the resident processors for acceleration of the AES algorithm.

The module includes a single object file that links to an instance of the BoringSSL library at build-time. This procedure creates a library that can then be linked to a calling application. The module performs no communications other than with the calling application (the process that invokes the module services) and the host operating system.

The module was tested and found to be compliant with FIPS 140-2 requirements on the platforms and environments listed in Table 2.

Table 2 – Tested Configurations

Operating System	Processor	Platform
Android 8.1 (32/64-bit)	Qualcomm Snapdragon 835	• Poly G7500 • Poly Studio X50
Android 9 (32/64-bit)	NXP i.MX 8M	• Poly CCX 700 • Poly Trio C60 • Poly TC8

[1] N/A – Not applicable

[2] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

The vendor affirms the module's continued validation compliance when operating on the following platforms[3]:

- Poly Studio X30
- Poly CCX 500
- Poly CCX 600

Additionally, per section G.5 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, the cryptographic module maintains validation compliance when the module and one of the identified unchanged tested operating systems are ported together to an untested platform. Note that if porting to an untested platform, there is no assurance of the minimum strength of generated keys.

2.2.1 Physical Cryptographic Boundary

As a firmware-hybrid, the module consists of disjoint firmware and hardware components within the same physical boundary of the host device. The firmware component of the module is a single object file (*bcm_object.o*). The hardware component is the CPU (Qualcomm's Snapdragon 835 processor or NXP's i.MX 8M processor) running on each host device. These components are entirely contained within the enclosure of each host device on which it is installed. The device enclosures comprise the module's physical cryptographic boundary.

2.2.2 Logical Cryptographic Boundary

The logical cryptographic boundary consists of all functionality contained within the module's compiled source code. Logically, the cryptographic boundary is the contiguous perimeter that surrounds all memory-mapped functionality provided by the module when loaded and stored in the host device's memory.

Figure 1 below shows the logical block diagram of the module executing in memory and its interactions with surrounding firmware components, as well as the module's physical and logical cryptographic boundaries.

[3] The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment not listed on the validation certificate.

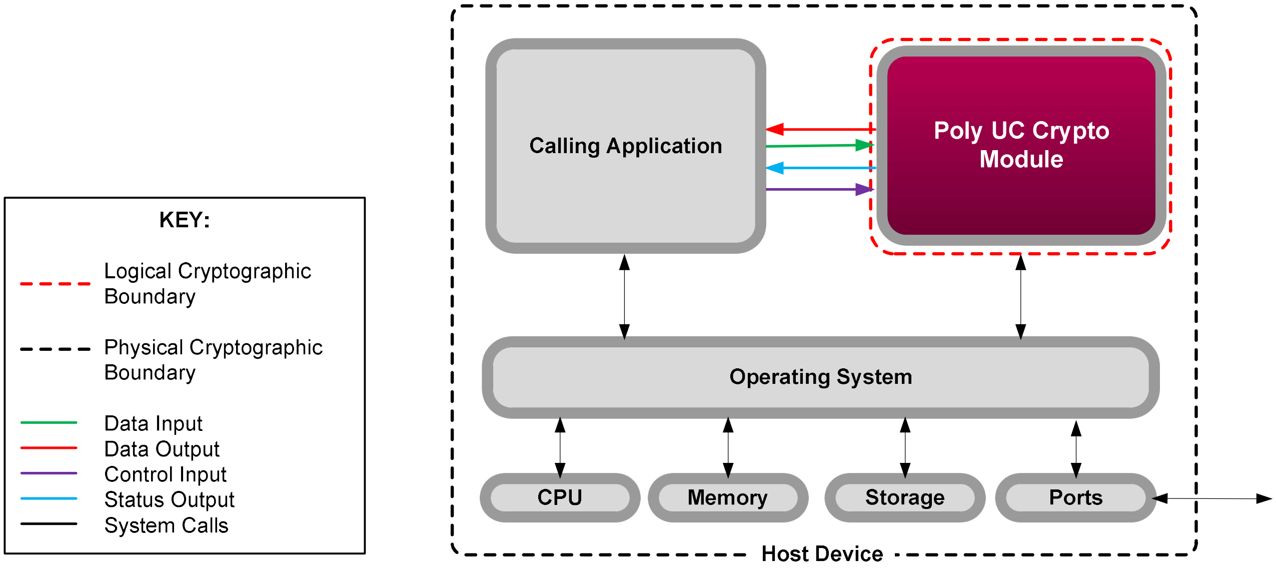

Figure 1 – Module Block Diagram (with Cryptographic Boundaries)

2.2.3 Algorithm Implementations

The module implements the FIPS-Approved algorithms listed in Table 3 below. The AES algorithm was performed leveraging the NEON extensions on the Qualcomm and NXP processors to provide PAA[4].

Table 3 – FIPS-Approved Cryptographic Algorithms

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1728	AES[5]	FIPS PUB 197 NIST SP 800-38A	CBC[6], ECB[7], CTR[8]	128, 192, 256	encryption/decryption
		FIPS PUB 197 NIST SP 800-38D	GCM[9]	128, 192, 256	encryption/decryption
		FIPS PUB 197 NIST SP 800-38D	GMAC[10]	128, 192, 256	MAC generation/verification
Vendor Affirmed	CKG[11]	NIST SP 800-133rev2	-	-	symmetric key generation
C1728	CVL	NIST SP 800-135rev1	TLS 1.0/1.1, 1.2	-	key derivation

[4] PAA – Processor Algorithm Acceleration
[5] AES – Advance Encryption Standard
[6] CBC – Cipher Block Chaining
[7] ECB – Electronic Code Book
[8] CTR – Counter
[9] GCM – Galois Counter Mode
[10] GMAC – Galois Message Authentication Code
[11] CKG – Cryptographic Key Generation

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1728	DRBG[12]	NIST SP 800-90Arev1	CTR-based	256-bit AES	deterministic random bit generation
C1728	ECDSA[13]	FIPS PUB 186-4	-	P-224, P-256, P-384, P-521	key pair generation
			-	P-224, P-256, P-384, P-521	public key validation
			SHA2-224, SHA2-256, SHA2-384, SHA2-512	P-224, P-256, P-384, P-521	digital signature generation
			SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	P-224, P-256, P-384, P-521	digital signature verification *This function is not available in the Approved mode of operation.*
C1728	HMAC[14]	FIPS PUB 198-1	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	KS<BS, KS=BS, KS>BS	message authentication
C1728	KTS[15]	FIPS PUB 197 NIST SP 800-38F	AES KW[16], AES KWP[17]	128, 192, 256	wrapping/unwrapping
C1728	RSA[18]	FIPS PUB 186-4	-	2048, 3072	key pair generation
			PKCS1.5	2048, 3072 (SHA2-224, SHA2-256, SHA2-384, SHA2-512)	digital signature generation
				1024, 2048, 3072 (SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512)	digital signature verification
			PSS	2048, 3072 (SHA2-224, SHA2-256, SHA2-384, SHA2-512)	digital signature generation
				2048, 3072 (SHA2-224, SHA2-256, SHA2-384, SHA2-512)	digital signature verification
C1728	SHS[19]	FIPS PUB 180-4	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	-	message digest
C1728	Triple-DES[20]	NIST SP 800-67rev2	CBC, ECB	Keying option 1	decryption[21]

***NOTE**: The module implements algorithms, modes, and keys that have been CAVS-tested but are not used by the module. Unless indicated otherwise, only those algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by the module in the Approved mode.*

[12] DBRG – Deterministic Random Bit Generator
[13] ECDSA – Elliptic Curve Digital Signature Algorithm
[14] HMAC – (keyed-) Hashed Message Authentication Code
[15] KTS – Key Transport Scheme
[16] KW – Key Wrap
[17] KWP – Key Wrap with Padding
[18] RSA – Rivest Shamir Adleman
[19] SHS – Secure Hash Standard
[20] DES – Data Encryption Standard
[21] Triple-DES encryption was CAVP-tested but is used solely to support the Triple-DES known answer self-test. Triple-DES encryption is not used as part of any other Approved/allowed service.

The vendor affirms the following cryptographic security methods:

- Cryptographic key generation – As per *NIST SP 800-133rev2*, the module uses its FIPS-Approved DRBG to generate cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG. The module's DRBG is seeded via entropy generated from a CPU jitter-based NDRNG[22], which is external to the module.

The module implements the non-Approved but allowed algorithms shown in Table 4 below.

Table 4 – Allowed Algorithms

Algorithm	Caveat	Use
MD5	-	TLS 1.0/1.1 handshake
NDRNG	-	seeding the module's DRBG

The module employs the following non-Approved algorithms (use of these algorithms is restricted to the module's non-Approved mode of operation):

- AES (non-compliant when using OFB, CFB, or CFB8 mode)
- ChaCha
- DES
- ECDH (non-compliant)
- ECDSA (non-compliant with non-Approved/untested functions and curves)
- MD4
- MD5 (when not used with TLS 1.0/1.1 operations)
- POLYVAL
- RSA (non-compliant with non-Approved/untested functions and key sizes)
- Triple-DES (non-compliant when used for encryption or with non-Approved keying options)

2.2.4 Modes of Operation

The module supports two modes of operation: Approved and Non-approved. The module will be in FIPS-Approved mode when all power up self-tests have completed successfully, and only Approved algorithms are invoked. See Table 3 and Table 4 above for a list of the Approved and allowed algorithms (respectively).

The module is designed and built by Plantronics developers to run in the Approved mode of operation by default (details regarding build/configuration steps have been omitted from this document, as they are outside the scope of this Security Policy). However, the module can alternate service-by-service between Approved and non-Approved modes of operation. The module will switch to the non-Approved mode upon execution of a non-Approved service. The module will switch back to the Approved mode upon execution of an Approved service. The module operator is responsible for zeroizing all keys when switching modes.

The module supports the Crypto Officer and User roles while in the non-Approved mode of operation.

[22] NDRNG – Non-Deterministic Random Number Generator

Section 2.4.2 below lists the services available in the non-Approved mode of operation.

2.3 Module Interfaces

The module isolates communications to logical interfaces that are defined in the firmware as an API[23]. The API is mapped to the following four logical interfaces:

- Data Input
- Data Output
- Control Input
- Status Output

The module's physical boundary features the physical ports of a host device. The module's manual controls, physical indicators, and physical, logical, and electrical characteristics are those of the host server. The module's logical interfaces are at a lower level in the firmware. The physical data and control input through physical mechanisms is translated into the logical data and control inputs for the firmware-hybrid module.

A mapping of the FIPS-defined interfaces, the host device's physical interfaces, and the module's logical interfaces can be found in Table 5. As required by section 1.9 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, all status and control ports and interfaces of the hybrid cryptographic module are directed through the firmware component's logical interfaces.

Table 5 – FIPS 140-2 Logical Interface Mappings

FIPS Interface	Physical Interface	Logical Interface
Data Input	Physical ports of the tested platforms	API input arguments that provide input data for processing
Data Output	Physical ports of the tested platforms	API output arguments that return generated or processed data back to the caller
Control Input	Physical ports of the tested platforms	API input arguments that are used to initialize and control the operation of the module
Status Output	Physical ports of the tested platforms	API call return values
Power Input	Physical ports of the tested platforms	N/A

2.4 Roles, Services, and Authentication

The sections below describe the module's authorized roles, services, and operator authentication methods.

[23] API – Application Programming Interface

2.4.1 Authorized Roles

There are two authorized roles that module operators may assume: Crypto Officer (CO) role and a User role. As per section 6.1 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, the calling application that loaded the module is its only operator. The module does not allow multiple concurrent operators.

The module supports the Crypto Officer and User roles while in the non-Approved mode of operation.

2.4.2 Operator Services

Descriptions of the services available are provided in Table 6 below. Please note that the keys and Critical Security Parameters (CSPs) listed in the table indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

Table 6 – Mapping of Operator Services to Inputs, Outputs, CSPs, and Type of Access

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Initialize	✓		Perform initialization of the module	API call parameters	Status	None
Run self-test on demand	✓	✓	Perform power-up self-tests	API call parameters	Status	None
Show status	✓	✓	Return the current mode of the module	API call parameters	Status	None
Zeroize	✓	✓	Zeroize and de-allocates memory containing sensitive data	Reboot; power cycle; unload module	None	All CSPs – W
Generate random number	✓	✓	Return random bits to the calling application	API call parameters	Status, random bits	DRBG seed – WRX DRBG entropy input – RX DRBG 'V' value – WRX DRBG 'Key' value – WRX
Generate message digest	✓	✓	Compute a message digest	API call parameters, message	Status, hash	None
Generate keyed hash	✓	✓	Compute a message authentication code	API call parameters, key, message	Status, MAC	HMAC key – RX
Perform symmetric encryption	✓	✓	Encrypt plaintext data	API call parameters, key, plaintext	Status, ciphertext	AES key – RX AES GMAC key – RX AES GCM key – RX AES GCM IV – RX

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Perform symmetric decryption	✓	✓	Decrypt ciphertext data	API call parameters, key, ciphertext	Status, plaintext	AES key – RX AES GMAC key – RX AES GCM key – RX AES GCM IV – RX Triple-DES key – RX
Generate asymmetric key pair	✓	✓	Generate a public/private key pair	API call parameters	Status, key pair	RSA public key – W RSA private key – W ECDSA public key – W ECDSA private key – W DRBG seed – WRX DRBG entropy input – RX DRBG 'V' value – WRX DRBG 'Key' value – WRX
Verify asymmetric key pair	✓	✓	Verify a public/private key pair	API call parameters	Status, key pair	RSA public key – RX RSA private key – RX ECDSA public key – RX ECDSA private key – RX
Generate signature	✓	✓	Generate a digital signature	API call parameters, key, message	Status, signature	RSA private key – RX ECDSA private key – RX DRBG seed – WRX DRBG entropy input – RX DRBG 'V' value – WRX DRBG 'Key' value – WRX
Verify signature	✓	✓	Verify a digital signature	API call parameters, key, signature, message	Status	RSA public key – RX
Perform key wrap	✓	✓	Perform key wrap function	API call parameters, wrapping key, key	Status, wrapped key	AES key wrap key – RX
Perform key unwrap	✓	✓	Perform key unwrap function	API call parameters, wrapping key, key	Status, unwrapped key	AES key wrap key – RX
Perform TLS key derivation	✓	✓	Perform key derivation function	API call parameters	Status, derived key	TLS master secret – X AES key – W AES GCM key – W

Table 7 below lists the services available in the non-Approved mode of operation.

Table 7 – Non-Approved Services

Service	Operator		Security Function(s)
	CO	User	
Generate message digest	✓	✓	MD4 MD5 POLYVAL

Service	Operator		Security Function(s)
	CO	User	
Perform symmetric encryption	✓	✓	AES (non-compliant) ChaCha DES Triple-DES (non-compliant)
Perform symmetric decryption	✓	✓	AES (non-compliant) ChaCha DES Triple-DES (non-compliant)
Generate signature	✓	✓	ECDSA (non-compliant) RSA (non-compliant)
Verify signature	✓	✓	ECDSA (non-compliant) RSA (non-compliant)
Perform key agreement	✓	✓	ECDH ECDSA (non-compliant)
Perform key transport	✓	✓	RSA (non-compliant)
Perform key generation	✓	✓	ECDSA (non-compliant) RSA (non-compliant)

2.4.3 Authentication

The module does not support authentication mechanisms; all operator roles are assumed implicitly.

2.5 Physical Security

As a multi-chip standalone firmware-hybrid module, the module relies on the host platforms' hardware to provide the mechanisms necessary to meet FIPS 140-2 level 1 physical security requirements. All components of the hardware are made of production-grade materials, and all integrated circuits are coated with commercial standard passivation.

2.6 Operational Environment

The module employs a non-modifiable operational environment. The operating system offers no mechanism whereby the operator can modify software/firmware components, nor can the operator load and execute software or firmware that was not included as part of the validation of the module.

2.7 Cryptographic Key Management

Entropy provided by the NXP i.MX 8M processor for DRBG input has a min-entropy value of 0.8975. Entropy provided by the Qualcomm Snapdragon 835 processor has a min-entropy value of 0.8300. These processors provide more than 256 bit of entropy per call, which is sufficient to support the apparent key strength of the generated keys.

The module supports the CSPs listed below in Table 8.

Table 8 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
AES key	128, 192, 256-bit AES CBC, ECB, CTR key	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Encryption, decryption
AES key wrap key	128, 192, 256-bit AES key	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Key wrap/unwrap
AES GMAC key	128, 192, 256-bit AES GMAC key	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	MAC generation
AES GCM key	128, 192, 256-bit AES GCM key	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Encryption, decryption
AES GCM IV[24]	96-bit value	Internally generated per scenario #2[25] of *FIPS 140-2 IG A.5*	Never	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Initialization vector for AES GCM
Triple-DES key	168-bit Triple-DES CBC, ECB key (Keying option 1)	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Decryption
HMAC key	Minimum 112-bit (variable) HMAC key	Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Message authentication with SHS
RSA private key	2048, 3072-bit RSA key	Internally generated via Approved DRBG OR Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Signature generation

[24] IV – Initialization Vector

[25] The IV generation method uses an Approved DRBG (seeded within the module's physical boundary) to generate a 96-bit IV and is in compliance with section 8.2.2 of *NIST SP 800-38D*.

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
RSA public key	1024, 2048, 3072-bit RSA key	Internally generated via Approved DRBG OR Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Signature verification *1024-bit keys used for signature verification only*
ECDSA private key	P-Curves P-224, P-256, P-384, P-521	Internally generated via Approved DRBG OR Input in plaintext via API call parameter	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Signature generation
ECDSA public key	P-Curves P-224, P-256, P-384, P-521	Internally generated via Approved DRBG	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Provided for use by calling application
TLS master secret	384-bit shared secret	Internally derived via TLS KDF	Output in plaintext	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Derivation of TLS session keys
DRBG entropy input	384-bit value	Externally generated[26] and input in plaintext via API call parameter	Never	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Entropy material for SP 800-90A DRBGs
DRBG seed	384 bits of random data	Internally generated using nonce along with DRBG entropy input	Never	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Seeding material for SP 800-90A DRBGs
DRBG 'V' value	128-bit internal state value	Internally generated	Never	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Used for CTR_DRBG
DRBG 'Key' value	256-bit internal state value	Internally generated	Never	Keys are not persistently stored by the module	Unload the module from memory; reboot or power-cycle the host device	Used for CTR_DRBG

[26] The module employs a non-deterministic random number generator which is outside of the logical cryptographic boundary.

2.8 EMI / EMC

The Poly UC Crypto Module was tested on the devices listed in Table 2 above. Table 9 lists the class conformances demonstrated by these devices when tested to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices.

Table 9 – FCC Part 15 Subpart B Class Conformance

Platform	Class Conformance
Poly Studio X50	A
Poly CCX 700	B
Poly Trio C60	B
Poly G7500	A
Poly TC8	B

2.9 Self-Tests

Self-tests are performed by the module when the module is first powered up and initialized, as well as during module operation when certain conditions exist. The following sections list the self-tests performed by the module, their expected error status, and the error resolutions.

2.9.1 Power-Up Self-Tests

The module performs the following self-tests at power-up:

- Firmware integrity check (using HMAC SHA2-256)
- AES-CBC encrypt and decrypt KATs[27] (128-bit)
- AES-GCM encrypt and decrypt KATs (128-bit)
- Triple-DES encrypt and decrypt KATs (168-bit)
- SHA-1/SHA2-256/SHA2-512 KATs
- DRBG instantiate/generate and reseed/generate KATs (256-bit AES)
- RSA sign/verify KAT (2048-bit)
- ECDSA sign KAT (curve P-256)

The module employs an HMAC SHA2-256 for the firmware integrity check. Thus, per section 9.3 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, no independent KAT is required for the HMAC or SHA2-256 implementations.

2.9.2 Conditional Self-Tests

The module performs the following conditional self-tests:

[27] KAT – Known Answer Test

- CRNGT[28] for the NDRNG
- RSA PCT[29]
- ECDSA PCT

2.9.3 Critical Functions Self-Tests

The module performs health checks for the DRBG's Generate, Instantiate, and Reseed functions as specified in section 11.3 of *NIST SP 800-90Arev1*. These tests are performed conditionally.

2.9.4 Self-Test Failure Handling

If any of the module's power-up self-tests fails, the module will enter a critical error state, abort the initialization process, and terminate execution, after which no cryptographic services will be accessible. If the CRNGT fails, the module will enter a critical error state and terminate execution, after which no cryptographic services will be accessible. The module can only recover from the critical error state by reinitializing the module (via restarting the calling application or rebooting the host device) and passing all power-on self-tests.

If the RSA PCT or the ECDSA PCT fails, the module will enter a soft error state. In this state, the module will report the error and immediately clear the error condition, allowing for the continued operation of the module.

If these recovery methods do not result in the successful completion of the self-tests, then the module will not be able to resume normal operations, and the CO should contact Plantronics, Inc. for assistance.

2.10 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

[28] RNG – Continuous Random Number Generator Test
[29] PCT – Pairwise Consistency Test

3. Secure Operation

The sections below describe how to ensure the module is operating in its validated configuration. **Operating the module without following the guidance herein (including the use of undocumented services) will result in non-compliant behavior and is outside the scope of this Security Policy**.

3.1 Module Setup

The following paragraphs describe the steps necessary to ensure that the Poly UC Crypto Module is running in its validated configuration.

3.1.1 Installation

The Poly Unified Communications Cryptographic Module is not delivered to end-users as a standalone offering or in source code form. Rather, it is packaged and distributed solely as an integrated component of Plantronics' Poly Unified Communications Software. Plantronics does not provide end-users with any mechanisms to directly access the module, its APIs, or any information sent to/from Poly UC software. The module is factory-installed with the Poly UC Software onto the target hardware platforms; no installation steps need to be performed by end-users.

3.1.2 Initialization

This module is designed to support Plantronics applications, and these applications are the sole consumers of the cryptographic services provided by the module. No end-user action is required to initialize the module for operation; the calling applications perform all initialization actions required to place the module into FIPS mode. The power-up integrity test and cryptographic algorithm self-tests are then performed automatically via a default entry point (DEP) when the module is loaded for execution by the calling application, without any specific action from the calling application or the end-user. End-users have no means to short-circuit or bypass these actions. Failure of any of the initialization actions will result in a failure of the module to load for execution.

3.1.3 Configuration

No configuration steps are required to be performed by end-users.

3.2 Operator Guidance

The following sections provide guidance to module operators for the correct and secure operation of the module.

3.2.1 Crypto Officer Guidance

No specific management activities are required to ensure that the module runs securely. However, if any irregular activity is noticed or the module is consistently reporting errors, then Plantronics Customer Support should be contacted.

3.2.2 User Guidance

Although the User does not have any ability to modify the configuration of the module, they should notify the CO if any irregular activity is noticed.

3.2.3 General Operator Guidance

The following provide further guidance for the general operation of the module:

- As the module supports service-by-service mode switching, the module's current mode of operation is determined by the service being executed at any given time. Execution of an Approved or allowed service means that the module is in Approved mode; execution of a non-Approved service means that the module is in non-Approved mode.

- To execute the module's power-up self-tests on-demand, the module's host device can be rebooted or power-cycled.

- As the module does not persistently store keys, the calling application is responsible for the storage and zeroization of keys and CSPs passed into and out of the module. The operating system and the calling application are responsible for the clean-up of any temporary and ephemeral keys. For the zeroization of keys in volatile memory, module operators can unload the module from memory or reboot/power-cycle the host device.

3.3 Additional Guidance and Usage Policies

The notes below provide additional guidance and policies that must be followed by module operators:

- The cryptographic module's services are designed to be provided to a calling application. Excluding the use of the NIST-defined elliptic curves as trusted third-party domain parameters, all other assurances from FIPS 186-4 (including those required of the intended signatory and the signature verifier) are outside the scope of the module and are the responsibility of the calling application.

- The calling application shall use entropy sources that meet the security strength required for the CTR_DRBG as shown in *NIST SP 800-90Arev1*, Table 3.

- Per FIPS 140-2 IG A.5, scenario #2, the AES-GCM IV is generated internally in its entirety randomly by the module's Approved DRBG. The DRBG seed is generated inside the module's physical boundary. The IV is 96 bits in length per NIST SP 800-38D, Section 8.2.2.

  In Approved mode, module operators shall not utilize GCM with an externally generated IV.

- In the event that the module encounters a DRBG self-test failure, the calling application must uninstantiate and re-instantiate the DRBG per the requirements found *in NIST SP 800-90Arev1*.

4. Acronyms

Table 10 provides definitions for the acronyms used in this document.

Table 10 – Acronyms

Acronym	Definition
AC	Alternating Current
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit
CRNGT	Continuous Random Number Generator Test
CSP	Critical Security Parameter
CTR	Counter
CVL	Component Validation List
DEP	Default Entry Point
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
EC	Elliptic Curve
ECB	Electronic Code Book
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI/EMC	Electromagnetic Interference /Electromagnetic Compatibility
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
HMAC	(keyed-) Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
NDRNG	Non-Deterministic Random Number Generator

Acronym	Definition
NIST	National Institute of Standards and Technology
OS	Operating System
PAA	Processor Algorithm Acceleration
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standard
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication
TDES	Triple Data Encryption Standard

Prepared by:
Corsec Security, Inc.

12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com