FIPS 140-2 Non-Proprietary Security Policy

Persistent Systems Wave Relay® Embedded Module

Level 1 Validation

Document Version 1.3

April 24, 2018

Prepared By:

Persistent Systems, LLC
303 Fifth Avenue Suite 306
New York, NY 10016
[www.persistentsystems. com](http://www.persistentsystems.com)

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the Wave Relay® Embedded Module .

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 products to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Communications Security Establishment (CSE) Cryptographic Module Validation Program (CMVP) owns the FIPS 140 program. The CMVP accredits independent testing labs to perform FIPS 140 testing; the CMVP also validates test reports for all products pursuing FIPS 140 validation. *Validation* 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 https://csrc.nist.gov/projects/testing-laboratories

1.2 About this Document

This non-proprietary Cryptographic Module Security Policy for the Wave Relay® Embedded Module 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 modules in a FIPS 140-2 mode of operation.

The Wave Relay® Embedded Module may also be referred to as the "module" in this document.

1.3 External Resources

The Persistent Systems website (http://www.persistentsystems.com) contains information on the full line of products from Persistent Systems, including a detailed overview of the Wave Relay® Embedded Module solutions. The Cryptographic Module Validation Program website (https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program) contains links to the FIPS 140-2 certificate and Persistent Systems contact information. The Cryptographic Module Validation Program website (https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program) contains links to the FIPS 140-2 certificate and Persistent Systems 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
DRBG	Deterministic Random Number Generator
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DTR	Derived Testing Requirement
FIPS	Federal Information Processing Standard
FTL	Flash Translation Layer
GUI	Graphical User Interface
HMAC	Keyed-Hash Message Authentication Code
KAT	Known Answer Test
MANET	Mobile Ad-hoc Network
MPU	Man Portable Unit
NIST	National Institute of Standards and Technology
NDRNG	Non Deterministic Random Number Generator
PCT	Pairwise Consistency Test
SHA	Secure Hashing Algorithm
TLS	Transport Layer Security

Table 1 – Acronyms and Terms

2 Persistent Systems Wave Relay® Embedded Module

2.1 Wave Relay® Product Overview

The Wave Relay® System is a peer-to-peer wireless MANET networking solution in which there is no master node. If any device fails, the rest of the devices continue to communicate using any remaining connectivity. By eliminating master nodes, gateways, access points, and central coordinators from the design, Wave Relay® delivers high levels of fault tolerance regardless of which nodes might fail. The system is designed to maximize the capacity of the radio frequency (RF) spectrum and to minimize the network overhead. While optimizing efficiency, Wave Relay® also implements techniques that increase multicast reliability. The advanced multicast functionality allows the system to support both multicast voice and video over IP.

Wave Relay® is designed to maintain high bandwidth connectivity among devices that are on the move. The system is scalable, enabling it to incorporate unlimited meshed devices into the wireless network, where the devices themselves form the communication infrastructure. Even in highly dynamic environments, the system is able to maintain connectivity by rapidly re-routing data as necessary. Wave Relay® is a self-forming and self-healing network where nodes can move freely within the network. Critical information flows reliably throughout the network while individual data paths are able to adapt at sub-second intervals. This unique approach creates an ideal environment for maximizing performance across the available communications medium. Customers leverage Wave Relay®'s straight forward and effective architecture to enable a true "Plug and Play" capability. Deploying a Wave Relay® network is as simple as connecting a standard Ethernet cable; customers are immediately connected to everything on the network.

Wave Relay® is a seamless wireless networking system offering a dynamic and reliable solution for all mobile networking needs The Persistent Systems Wave Relay Embedded Module offers the Wave Relay® MANET combined with other leading-edge technologies in a single smart radio.

2.2 Cryptographic Module Specification

The module is the Wave Relay® Embedded Module HW P/N WR-5200 Version 4.0

The module uses FW Version **19.3.1.** Each module is a multiple-chip embedded embodiment.

For the WR-5200, the physical cryptographic boundary is defined as the module board with heat-sinks, which includes the Wave Relay® main board, including the hardware cryptographic accelerator chip, drivers, CPU, and on-board flash memory. The boundary does not include the radio module or power supply.

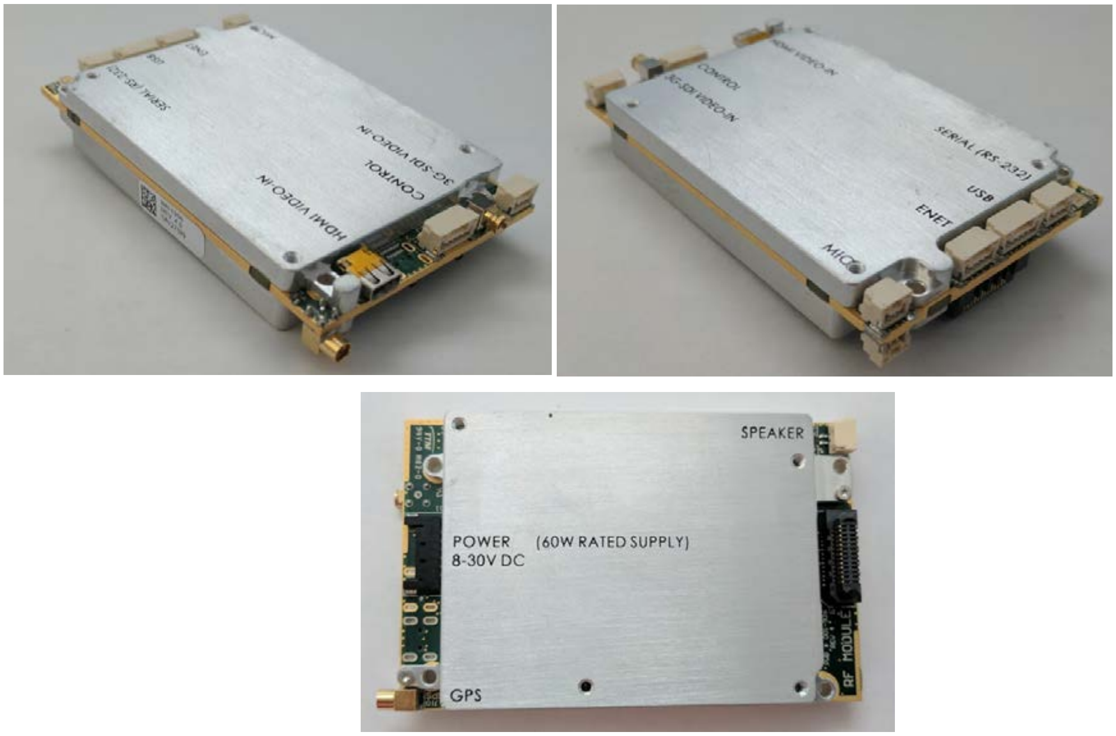

Figure 1 – Physical Boundary of Embedded Module

The module is in FIPS-approved mode of operation when the validated firmware is used and when only Approved/allowed functionality is used. It does not have any bypass capability.

2.2.1 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	2
Finite State Model	1
Physical Security	1
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A
Overall Level	**1**

Table 2 – Validation Level by DTR Section

2.2.2 Algorithm Implementation Certificates

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

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Symmetric Key	AES- {128*, 192*, 256} in {CBC*, CTR, ECB*} mode	FIPS 197	Cert. #4456	Data encryption / decryption
Keyed Hash	HMAC-SHA-{1, 224*, 256}	FIPS 198	Cert. #2957	Message integrity
Hashing	SHA- {1, 224*, 256}	FIPS 180-4	Cert. #3668	Message digest

Table 3 – Algorithm Certificates for Wave Relay® E2 Cryptographic Engine

* Denotes that the algorithm, mode of operation, and/or key size is not used/accessible

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Symmetric Key	AES- {128*, 192*, 256*} in {CBC*, CTR*, GCM*, ECB*} mode	FIPS 197	Cert. #4454	Not currently used/accessible
Symmetric Key	AES- {128*, 256*} in {XTS*} mode	FIPS 197	Cert. #4454	Not currently used/accessible
Keyed Hash	HMAC-SHA-{1*, 224*, 256*, 384*, 512}	FIPS 198	Cert. #2955	Message integrity
Hashing	SHA-{1*, 224*, 256*, 384*, 512}	FIPS 180-4	Cert. #3666	Message digest

Table 4 – Algorithm Certificates for Wave Relay® Cryptographic Kernel

* Denotes that the algorithm, mode of operation, and/or key size is not used/accessible

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Symmetric Key	AES- {128, 192, 256} in {CBC, OFB*, CTR, GCM[1], ECB*, CFB-1*, CFB-8*, CFB-128*} mode	FIPS 197	Cert. #4455	Data encryption / decryption
CKG	Cryptographic Key Generation: Asymmetric signature key generation using unmodified DRBG output Direct symmetric key generation using unmodified DRBG output Derivation of symmetric keys from a key agreement shared secret.	SP800-133	Vendor Affirmed	Key Generation
Asymmetric Key	ECDSA SigGen Component - 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*	SP 800-56A	CVL #1164	Not currently used/accessible
Transport Layer Security (TLS)	Section 4.2, TLS- TLS (TLS1.0/1.1 TLS1.2 (SHA 256, 384, 512)) No parts of this protocol, other than the KDF, have been tested by the CAVP and CMVP.	SP 800-135 Section 4.2	CVL #1163	Key Derivation
Asymmetric Key	RSA Decryption Primitive - RSADP: (Mod2048)	SP 800-56B Section 7.1.2	CVL #1162	Key Recovery
KAS ECC	ECC CDH - Curves: B-233*, B-283*, B-409*, B-571*, K-233*, K-283*, K-409*, K-571*, P-224*, P-256*, P-384*, P-521* KAS ECC – (EC: P-256 SHA256)* (ED: P-384 SHA384) (EE: P-521 SHA512)*	SP 800-56Are2 Section 5.7.1.2	CVL #1161	Key Agreement

[1] The module is compatible with TLSv1.2 and provides support for the acceptable GCM cipher suites from SP 800-52 Rev1, Section 3.3.1. 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, client or server, to encounter this condition will trigger a handshake to establish a new encryption key. In case the module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Deterministic Random Bit (DRBG)	CTR_DRBG - {128, 192, 256}-CTR*, HASH_DRBG - SHA- {1, 224, 256, 384, 512}*, HMAC_DRBG - SHA- {1*, 224*, 256*, 384*, 512}}	SP 800-90A	Cert. #1443	Random Bit Generation
Asymmetric Key	DSA - PQG(gen): (2048, 224) SHA(224, 256, 384, 512)* (2048, 256) SHA(256, 384, 512)* (3072, 256) SHA(256, 384, 512)* PQG(ver): (1024, 160) SHA(1 , 224, 256, 384, 512)* (2048, 224) SHA(224, 256, 384, 512)* (2048, 256) SHA(256, 384, 512)* (3072, 256) SHA(256, 384, 512)* KeyPairGen: (2048, 224)* (2048, 256)* (3072, 256)* SIG(gen): (2048, 224) SHA(224, 256, 384, 512)* (2048, 256) SHA(224, 256, 384, 512)* (3072, 256) SHA(224, 256, 384, 512)* SIG(ver): (1024, 160) SHA(1, 224, 256, 384, 512)* (2048, 224) SHA(1, 224, 256, 384, 512)* (2048, 256) SHA(1, 224, 256, 384, 512)* (3072, 256) SHA(1, 224, 256, 384, 512)*	FIPS 186-4	Cert. #1191	Not currently used/accessible

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Asymmetric Key	ECDSA Key Pair Gen, Sig Gen, Sig Ver 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*) Public Key Validation: CURVES (P-192* P-224* 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*) SigGen: CURVES (P-224*: (SHA-224*, 256*, 384*, 512*) P-256*: (SHA-224*, 256*, 384*, 512*) P-384: (SHA-224*, 256*, 384, 512*) P-521*: (SHA-224*, 256*, 384*, 512*) K-233*: (SHA-224*, 256*, 384*, 512*) K-283*: (SHA-224*, 256*, 384*, 512*) K-409*: (SHA-224*, 256*, 384*, 512*) K-571*: (SHA-224*, 256*, 512*) B-233*: (SHA-224*, 256*, 384*, 512*) B-283*: (SHA-224*, 256*, 384*, 512*) B-409*: (SHA-224*, 256*, 384*, 512*) B-571*: (SHA-224*, 256*, 384*, 512*)) SigVer: CURVES (P-192*: (SHA-1*, 224*, 256*, 384*, 512*) P-224*: (SHA-1*, 224*, 256*, 384*, 512*) P-256*: (SHA-1*, 224*, 256*, 384*, 512*) P-384: (SHA-1*, 224*, 256*, 384, 512*) P-521*: (SHA-1*, 224*, 256*, 384*, 512*) K-233*: (SHA-1*, 224*, 256*, 384*, 512*) K-283*: (SHA-1*, 224*, 256*, 384*, 512*) K-409*: (SHA-1*, 224*, 256*, 384*, 512*) K-571*: (SHA-1*, 224*, 256*, 384*, 512*) B-233*: (SHA-1*, 224*, 256*, 384*, 512*) B-283*: (SHA-1*, 224*, 256*, 384*, 512*) B-409*: (SHA-1*, 224*, 256*, 384*, 512*) B-571*: (SHA-1*, 224*, 256*, 384*, 512*)) **Note: Only P-384 curve with SHA-384 is employed in this module**	FIPS 186-4	Cert. #1085	Signature Generation & Verification
Keyed Hash	HMAC-SHA-{1, 224*, 256, 384*, 512}	FIPS 198	Cert. #2956	Message integrity

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
KTS	AES Cert. #4455 and HMAC Cert. #2956; key establishment methodology provides between 128 and 256 bits of encryption strength)	IG D.9	Cert. #4455, #2956	Key Transport

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Asymmetric Key	RSA - 186-2 Sig Ver 9.31*: Modulus lengths (in bits): 1024, 1536, 2048, 3072, 4096 SHAs: SHA-{1, 256, 384, 512} Sig Ver PKCS1.5*: Modulus lengths (in bits): 1024, 1536, 2048, 3072, 4096 SHAs: SHA-{1, 256, 384, 512} Sig Ver PSS*: Modulus lengths (in bits): 1024, 1536, 2048, 3072, 4096 SHAs: SHA-{1, 256, 384, 512} RSA - 186-4 Key Gen 9.31: Public Key Exponent: Fixed Probable Random Primes: Mod lengths (in bits): 2048, 3072* Primality Tests: C.2 Sig Gen 9.31*: Mod 2048 SHA: SHA-{1, 256, 384, 512} Mod 3072 SHA: SHA-{1, 256, 384, 512} Sig Ver 9.31*: Mod 1024 SHA: SHA-{1, 256, 384, 512} Mod 2048 SHA: SHA-{1, 256, 384, 512} Mod 3072 SHA: SHA-{1, 256, 384, 512} Sig Gen PKCS1.5: Mod 2048 SHA: SHA-{1*, 224*, 256, 384*, 512*} Mod 3072* SHA: SHA-{1*, 224*, 256*, 384*, 512*} Sig Ver PKCS1.5: Mod 1024* SHA: SHA-{1*, 224*, 256*, 384*, 512*} Mod 2048 SHA: SHA-{1*, 224*, 256, 384*, 512} Mod 3072* SHA: SHA-{1*, 224*, 256*, 384*, 512*} Sig Gen PSS*: Mod 2048 SHA: SHA-{1, 224, 256, 384, 512} Mod 3072 SHA: SHA-{1, 224, 256, 384, 512} Sig Ver PSS*: Mod 1024 SHA: SHA-{1, 224, 256, 384, 512} Mod 2048 SHA: SHA-{1, 224, 256, 384, 512} Mod 3072 SHA: SHA-{1, 224, 256, 384, 512}	FIPS 186-2 FIPS 186-4	Cert. #2433	Key Generation Signature Generation & Verification

Algorithm Type	Algorithm	Standard	CAVP Cert.	Use
Hashing	SHA-{1, 224*, 256, 384, 512}	FIPS 180-4	Cert. #3667	Message digest

Table 5 – Algorithm Certificates for Wave Relay® Cryptographic Library

* Denotes that the algorithm, mode of operation, and/or key size is not used/accessible

The following non-approved, but allowed protocols/algorithms are available in FIPS mode of operation:

- EC Diffie-Hellman (CVL Cert. #1161, key agreement; key establishment methodology provides 192 bits of encryption strength)
- MD5 within TLS only*
- Hardware non-deterministic random number generator (NDRNG) (allowed for seeding FIPS-approved DRBG)
- RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength)

* No security is claimed from the use of these protocols/algorithms.

2.3 Module Interfaces

The interfaces for the cryptographic boundary include physical and logical interfaces. The physical interfaces provided by the module are mapped to five FIPS 140-2 defined logical interfaces: Data Input, Data Output, Control Input, Status Output, and Power. The mapping of logical interfaces to module physical interfaces is provided in the following table:

Module Physical Interface Persistent Systems Wave Relay® Embedded Module	FIPS 140-2 Logical Interface
Port	
HDMI Video In	Data Input
3G-SDI Video In	Data Input
Serial (RS-232)	Data Input, Data Output, Control Input, Status Output
USB	Data Input, Data Output, Control Input, Status Output
Ethernet	Data Input, Data Output, Control Input, Status Output
Mic	Data Input
GPS	Data Input
Radio Data	Data Input, Data Output, Control Input, Status Output
Speaker	Data Output, Status Output
Power/Zeroize/GPIO Control	Control Input
Status LED	Status Output
Power Input Port	Power

Table 6 – Logical Interface / Physical Interface Mapping

2.4 Roles, Services, and Authentication

The module only supports a FIPS-Approved mode. The module is accessed via Web browser over HTTPS/TLS. As required by FIPS 140-2, the module supports a Crypto Officer role and a User role. The module supports role-based authentication, and the respective services for each role are described in the following sections.

Both roles can access all services in the module. The module does not support a Maintenance role. The "Unauthenticated" role indicates services that the module performs automatically after POST and services that an operator may perform without authentication.

2.4.1 Operator Services and Descriptions

The services available to roles in the modules are as follows:

Service	Description	Roles	CSPs
Power-On	Provides power and initializes the module. TLS key pairs are generated if unit was previously zeroized.	Unauthenticated	Use Module Integrity Key Use Store Key Use/Zeroize DRBG entropy input Use/Zeroize DRBG V Use/Zeroize DRBG Key Generate/Use/Zeroize CA Private Key Generate/Use/Zeroize CA Public Key Generate TLS ECDSA Private Key Generate TLS ECDSA Public Key Generate TLS RSA Private Key Generate TLS RSA Public Key Generate TLS Pre-master Secret (All Cases) Generate TLS Master Secret/Traffic Keys[2]
Packet Forwarding	Provides packet forwarding and receipt. Forwarded packets are encrypted and signed, and incoming packets are decrypted and verified	Crypto Officer User	Use MANET Encryption Key Use MANET Authentication Key
Management	Provides configuration and password management functions over TLS such as setting and deleting a password	Crypto Officer User	Write/Use/Zeroize Operator Passwords

[2] "Traffic Keys" refer to the MANET Encryption Key and MANET Authentication Key listed in Table 8.

Service	Description	Roles	CSPs
TLS	Establishes and maintains TLS connections	Crypto Officer User	Generate/Use/Zeroize DRBG entropy input Use/Zeroize DRBG V Use/Zeroize DRBG Key Generate/Use/Zeroize TLS Elliptic Curve Diffie-Hellman Private Key Generate/Use/Zeroize TLS Elliptic Curve Diffie-Hellman Public Key Use/Zeroize TLS Elliptic Curve Diffie-Hellman Shared Secret Use TLS ECDSA Private Key Use TLS ECDSA Public Key Use TLS RSA Private Key Use TLS RSA Public Key Use/Zeroize TLS Pre-master Secret Use/Zeroize TLS Master Secret/Traffic Keys
Manage MANET Keys	Generates MANET Encryption and Authentication Keys for encrypt/decrypt operations	Crypto Officer User	Read/Write/Generate/Zeroize MANET Encryption Key Read/Write/Generate/Zeroize MANET Authentication Key

Service	Description	Roles	CSPs
Firmware Upgrade	Upgrade firmware to newer release Note: If non-FIPS validated firmware is loaded, the module is no longer a FIPS validated module.	Crypto Officer User	Use Firmware Upgrade Public Key Use Firmware Decryption Key
Self-Test	Performs self-tests on critical functions of module	Crypto Officer User Unauthenticated	Use Module Integrity Key
Status	Status of the module	Crypto Officer User Unauthenticated	N/A
Zeroize	Zeroize keys and CSPs in the module	Crypto Officer User Unauthenticated	Zeroize Operator Passwords Zeroize MANET Encryption Key Zeroize MANET Authentication Key Zeroize TLS ECDSA Private Key Zeroize TLS ECDSA Public Key Zeroize TLS RSA Private Key Zeroize TLS RSA Public Key Zeroize Store Key
Application Loading	The operating system is modifiable and allows the operator to load and execute software that was not included as part of the original evaluation.	Crypto Officer User Unauthenticated	N/A

Table 7 – Operator Services and Descriptions

The module does not support multiple concurrent operators. Each “view” or “set” of configuration by a user is a separate action, and the actual configuration is determined by the latest “set.” The Web GUI will indicate that a User/Crypto Officer role has logged themselves in. Only one operator can configure the module at one time. In the event that two authenticated sessions exist at one time for configuration, the module will save/store the parameters of the last operation. Concurrent sessions are treated as an individual session, but from separate end points.

2.4.2 Operator Authentication

Crypto Officer and User password must be a minimum of 8 characters. Legal password characters are the set of all 95 printable ASCII characters. This includes a-z, A-Z, 0-9, space, and these special characters: `!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`. Passwords are case-sensitive. Given a random password of eight characters using the full character set, the probability of a successful random attempt is $1/95^8$, which is dramatically less than the 1/1,000,000 requirement. There is an explicit limit employed by the module to dramatically slow down the effective speed of an online brute force guessing attack. The system keeps tracks of recent failed attempts. If this count reaches ten, the system no longer accepts authentication attempts and the system reduces this count by one every ten seconds. As a result, a maximum of 16 guesses can be attempted in a one minute interval. This assumes that there are no failed guesses in the prior 100 seconds, ten guesses are made immediately at the beginning of the minute, and then followed by one guess every ten seconds for the remainder of the minute. Given a random password of eight characters using the full character set, this reduces the probability of success to $16/95^8$, which is dramatically less than 1/100,000 requirement.

2.5 Physical Security

The physical security of the cryptographic module meets FIPS 140-2 Level 1 requirements. The cryptographic module consists of production-grade components. The physical boundary of the cryptographic module is the same as the physical boundary of the device.

The module does not include a maintenance interface; therefore, the FIPS-140-2 maintenance mode requirements do not apply.

2.6 Operational Environment

The module supports a modifiable operational environment. The module's firmware can only be updated with the verification of a digital signature over the firmware to be loaded. In addition, the operator may load and execute software that was not included in the original evaluation as the underlying operational environment is modifiable.

2.7 Cryptographic Key Management

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

Key/CSP Name	Description / Use	Generation / Establishment	Storage	Import/Export	Destruction
MANET Encryption Key	AES CTR mode with 256-bit key for encryption / decryption of network traffic	Internal generation by DRBG or imported via TLS	**Storage**: Flash in encrypted form by the Store Key **Association**: The system is the one and only owner. Relationship is maintained by the operating environment via protected memory.	**Agreement**: NA **Entry**: Electronic Key Entry via TLS **Output**: via TLS	Destroyed by zeroizing the Store Key
MANET Authentication Key	Minimum key size of 256 bits. Maximum key size is the size of the block algorithm used. HMAC-SHA1 and HMAC-SHA256 has a block size of 512. HMAC-SHA512 has a block size of 1024 bits. This key is used for message verification and integrity check.	Internal generation by DRBG or imported via TLS	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: Electronic Entry via TLS **Output**: via TLS	Destroyed by zeroizing the Store Key
Module Integrity Key	HMAC SHA-256 key for verifying the integrity of the module. Fixed string of 43 characters.	Not generated by the module; built into firmware	**Storage**: Flash in plaintext **Type:** Static	**Agreement**: NA **Entry**: FW upgrade encrypted by Firmware Decryption Key and TLS **Output**: NA	Replaced during FW upgrade
Firmware Decryption Key	AES CTR 256-bit key for decryption of firmware before upgrade	Not generated by the module; built into firmware	**Storage**: Flash in plaintext **Type:** Static	**Agreement**: NA **Entry**: FW upgrade encrypted by itself and TLS **Output**: NA	Replaced during FW upgrade

Key/CSP Name	Description / Use	Generation / Establishment	Storage	Import/Export	Destruction
Firmware Upgrade Public Key	RSA 15360-bit key for verifying firmware signature before upgrading	Not generated by the module; built into firmware	**Storage**: Flash in plaintext **Type:** Static	**Agreement**: NA **Entry**: FW upgrade encrypted Firmware Decryption Key and TLS **Output**: NA	Replaced during FW upgrade
Operator Passwords	Alphanumeric passwords externally generated by a human user for authentication.	Not generated by the module; imported by the human operator	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: Electronic entry via TLS. **Output**: NA	Destroyed by zeroizing the Store Key
Store Key	AES CBC 256-bit key for encryption of Flash data store	Internal generation by DRBG	**Storage**: Flash (without FTL) in plaintext	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroize
DRBG entropy input	960-bits of input from the NDRNG. Expected entropy is significantly greater than 512 bits.	Hardware based entropy source used to construct seed	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
DRBG V	The DRBG V consists of 512-bits and is part of the internal state upon which the security of this DRBG mechanism depends.	Generated first during DRBG instantiation and then subsequently updated using the DRBG update function	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
DRBG Key	The DRBG Key consists of 512-bits and is part of the internal state upon which the security of this DRBG mechanism depends.	Generated first during DRBG instantiation and then subsequently updated using the DRBG update function	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use

Key/CSP Name	Description / Use	Generation / Establishment	Storage	Import/Export	Destruction
RSA CA Public Key	RSA Public 2048-bit certificate signature	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: via TLS	Destroyed by zeroizing the Store Key
RSA CA Private Key	RSA Private 2048-bit certificate signature	Internal generation by DRBG	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
ECDSA CA Public Key	ECDSA Public P-384 certificate signature	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: via TLS	Destroyed by zeroizing the Store Key
ECDSA CA Private Key	ECDSA Private P-384 certificate signature	Internal generation by DRBG	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
TLS Elliptic Curve Diffie-Hellman Shared Secret	The shared secret used in Elliptic Curve Diffie-Hellman (ECDH) exchange. The size of the shared secret is 384-bits.	Established per the ECDH key agreement	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
TLS Elliptic Curve Diffie-Hellman Private Key	The private key used in Elliptic Curve Diffie-Hellman (ECDH) exchange. Using the P-384 curve.	Internal generation by DRBG	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Zeroized after use
TLS Elliptic Curve Diffie-Hellman Public Key	The public key used in Elliptic Curve Diffie-Hellman (ECDH) exchange. Using the P-384 curve.	Internal generation by DRBG	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: Part of TLS handshake **Output**: Part of TLS handshake	Zeroized after use

Key/CSP Name	Description / Use	Generation / Establishment	Storage	Import/Export	Destruction
TLS ECDSA Private Key	Signature generation. Using the P-384 curve.	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: NA	Destroyed by zeroizing the Store Key
TLS ECDSA Public Key	Signature verification. Using the P-384 curve.	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: Part of TLS handshake	Destroyed by zeroizing the Store Key
TLS ECDSA Public Key of the remote server	To authenticate the TLS key agreement. Using the P-384 curve.	Received as part of the TLS handshake when using ECDSA cipher suites	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: Part of TLS handshake **Output**: NA	Automatically when TLS session is terminated
TLS RSA Private Key	Identity certificates used in TLS negotiations. 2048 bits in size.	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: NA	Destroyed by zeroizing the Store Key
TLS RSA Public Key	Identity certificates used in TLS negotiations. 2048 bits in size.	Internal generation by DRBG	**Storage**: Flash in encrypted form by the Store Key	**Agreement**: NA **Entry**: NA **Output**: Part of TLS handshake	Destroyed by zeroizing the Store Key
TLS RSA Public Key of the remote server	To encrypt the TLS Pre-master Secret using RSA Key Transport. 2048 bits in size.	Received as part of the TLS handshake when using RSA cipher suites	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: Part of TLS handshake **Output**: NA	Automatically when TLS session is terminated

Key/CSP Name	Description / Use	Generation / Establishment	Storage	Import/Export	Destruction
TLS Pre-master Secret	Used to derive the TLS Master Secret and session keys. 384 bits in size	Establishment depends on cipher suite used and client/server role. **RSA client**: Internal generation by DRBG **RSA server**: Decrypted using the TLS RSA Private Key **ECDH**: Established by ECDH Key Agreement	**Storage**: RAM in plaintext **Type:** Ephemeral	Import/Export depends on cipher suite used and client/server role. **RSA client:** Output encrypted by the RSA Public Key of the server via RSA Key Transport **RSA server**: Entered encrypted by the RSA Public Key of the server via RSA Key Transport **ECDH**: Elliptic Curve Diffie-Hellman Key Agreement during TLS handshake	Automatically when TLS session is terminated
TLS Master Secret	Used in TLS connections to derive session keys. 384 bits in size.	Established using TLS protocol. This key was derived in the module.	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Automatically when TLS session is terminated
TLS Encryption Key	AES 128, 192, or 256-bit keys in GCM or CBC mode. Used in TLS connections.	Established using TLS protocol. This key was derived in the module.	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Automatically when TLS session is terminated
TLS Integrity Key	HMAC-SHA-1. Used in TLS connections. 160 bits in size.	Established using TLS protocol. This key was derived in the module.	**Storage**: RAM in plaintext **Type:** Ephemeral	**Agreement**: NA **Entry**: NA **Output**: NA	Automatically when TLS session is terminated

Table 8 – Key/CSP Management Details (also includes public keys)

Network Keys can be exported from the physical boundary of the module when the Crypto Officer re-keys the module using the network management feature. The Network Key will be sent to other nodes (modules) on the network encrypted with TLS.

All persistent keys and CSPs are stored in an encrypted store. This store is located in eMMC and is encrypted via an AES 256-bit key (Store Key). The key & IV used to encrypt the store are stored in a separate flash without FTL. Zeroization has been implemented to ensure no traces are left of the store key & IV. Zeroization is achieved by explicitly erasing the flash sector, containing the key and IV material. The erase operation is at the hardware level and writes a specific value to flash. The Embedded Module can be zeroized by switching the zeroized pin, in the control port, to ground (requires main power to be connected to a power source), or via the management interface by an authorized role (requires unit to be on and operational).

2.8 Self-Tests

The module includes an array of self-tests that are run during startup and periodically during operations to prevent secure data from being released and to ensure all components are functioning correctly. In the event of any self-test failure, the module will restart. Self-test Success status is indicated by the status LED as well as via HTTPS. No keys or CSPs will be output when the module is in an error state.

If the self-tests succeed, the operator will be presented with a login screen when accessing the the module via HTTPS. Attempts to access it via HTTP will be automatically redirected to HTTPS. If the self-tests fail, any attempt to access the module will fail.

Since the module only supports a FIPS-approved mode of operation, the self-tests are always run. On failure, the module will always be non-operational as there is no non-FIPS or bypass mode available.

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

2.8.1 Power-On Self-Tests

Power-on self-tests are run upon every initialization of the module and if any of the tests fail, the process will be halted and the module will not initialize. In this error state, no services can be accessed by the users. The module implements the following power-on self-tests:

Hardware Implementation (Cryptographic Engine)	
Test Target	**Description**
AES	KATs: Encryption, Decryption Modes: CBC, CTR, ECB Key sizes: 128-bits, 192-bits, 256-bits
SHS	KATs: Output Verification SHA sizes: SHA-1, SHA-224, SHA-256
HMAC	KATs: Generation, Verification SHA sizes: SHA-1, SHA-224, SHA-256

Table 9 – Cryptographic Engine POST

Firmware Implementation (Cryptographic Kernel)	
Test Target	**Description**
AES	KATs: Encryption, Decryption Modes: CBC, CTR, ECB, GCM Key sizes: 128-bits, 192-bits, 256-bits
AES	KATs: Encryption, Decryption Mode: XTS Key sizes: 128-bits, 256-bits
SHS	KATs: Output Verification SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
HMAC	KATs: Generation, Verification SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512

Table 10 – Cryptographic Kernel POST

Firmware Implementation (Cryptographic Library)	
Test Target	**Description**
AES	KATs: Encryption, Decryption Modes: ECB Key sizes: 128-bits
AES	KATs: Encryption, Decryption Mode: XTS Key sizes: 128-bits, 256-bits
DSA	PCT: Signature Generation, Signature Verification Key size: 2048-bits
DRBG	KATs: HASH_DRBG, HMAC_DRBG, CTR_DRBG Security Strengths: 256-bits
ECDSA	PCTs: Key Generation, Signature Generation, Signature Verification Curves: P-224, K-233
ECC CDH	Shared secret calculation per SP 800-56A §5.7.1.2, IG 9.6 Curve: P-224
ECC KAS	KAT is performed by ECC CDH KAT
GCM	KATs: Generation, Verification Key sizes: 256-bits
RSA	KATs: Signature Generation, Signature Verification Key sizes: 2048-bits

Firmware Implementation (Cryptographic Library)	
Test Target	**Description**
SHS	KATs: Output Verification SHA sizes: SHA-1
HMAC	KATs: Generation, Verification SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512

Table 11 – Cryptographic Library POST

The module performs all power-on self-tests automatically when it is initialized. The module also verifies its integrity using HMAC-SHA256. Successful completion of self-tests will be indicated via HTTPS. All power-on self-tests must be passed before a User/Crypto Officer can perform services. The Power-on self-tests can be run on demand by restarting the module.

2.8.2 Conditional Self-Tests

Conditional self-tests are run continuously when certain conditions are met during operation of the module. The module performs the following conditional self-tests:

Conditional Self-Tests		
Test Target	**Condition**	**Description**
RSA	On each generation of a key pair	Pairwise consistency test
DSA	On each generation of a key pair	Pairwise consistency test
ECDSA	On each generation of a key pair	Pairwise consistency test
DRBG	On output of DRBG implementation	Continuous test
NDRNG	On output of NDRNG (seed for DRBG)	Continuous test
RSA digital signature verification	Firmware Load / Firmware Upgrade	Signature verification test
DRBG	SP800-90A Health Tests	Health Checks

Table 12 – Conditional Self-Tests

Note that the module performs conditional tests for firmware and software implementations of the algorithms listed in the Algorithm Implementation Certificates section. If any of these tests fail, the module will enter an error state. The module can be re-initialized to clear the error and resume FIPS mode of operation. While in an error state, no services can be accessed by the operators.

2.9 EMI/EMC

The module meets Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for a radio.

2.10 Mitigation of Other Attacks

The module does not mitigate other attacks.

3 Guidance and Secure Operation

This section describes how to configure the module for FIPS-approved mode of operation. Operating it without maintaining the following settings will violate the FIPS-approved mode of operation.

3.1 Crypto Officer and User Guidance

3.1.1 Initialization for FIPS Mode of Operation

The Crypto Officer or User must configure and enforce the following procedures:

1. When setting the password, the Crypto Officer or User must select passwords with a minimum length of eight legal characters, which is enforced by the module. Legal password characters are the set of all 95 printable ASCII characters. This includes a-z, A-Z, 0-9, space, and these special characters: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~.

   Note: Stronger, more secure passwords should have a combination of letters and numbers and should not contain any recognizable words that may be found in a dictionary. The module does not enforce this; the Crypto Officer or User must follow his/her organization's systems security policies and adhere to the password policies set forth therein.

2. Ensure FW version running is listed in section 2.2 of this document.

3.1.2 General Crypto Officer and User Guidance

After initialization for FIPS mode, the Crypto Officer and User should follow the guidance below:

1. When entering a network key over the configuration GUI, the operator must ensure the key was generated by FIPS-approved methods and that the key was not previously used.
2. The operator must ensure that all Radio MAC addresses used in a network are unique.
3. The Crypto Officer or User must not disclose passwords and must store passwords in a safe location and according to his/her organization's systems security policies for password storage.