DRAEGER WCM9113 802.11ABGN VG2 FIPS 140-2 Non-Proprietary Security Policy

Table of Contents

Table of Figures

Table of Tables

1 Introduction

The DRAEGER WCM9113 802.11ABGN VG2 herein after referred to as “cryptographic module” or “module”, (Part Number: MS32018 Rev. 03; FW Version: 1.8.2 with bootloaders 1.7 and 1.8) is a FIPS 140-2 Level 1 multi-chip embedded cryptographic module that has been licensed to Draeger Medical Systems, Inc. for exclusive use by Redpine Signals, Inc.

2 Cryptographic Boundary

The cryptographic boundary is defined as the outer perimeter of a production grade multi-chip embedded printed circuit board that includes a cryptographic processor, RF and peripheral memory which are contained within the production grade metal enclosure.

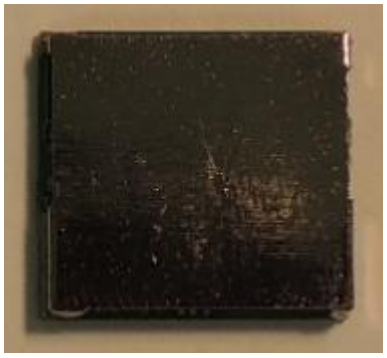

Figure 1: Top Side of the Cryptographic Module

Figure 2: Bottom Side of the Cryptographic Module

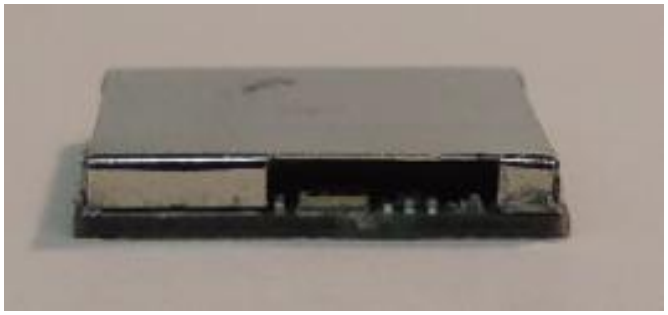

Figure 3: Front Side of the Cryptographic Module

Figure 4: Right Side of the Cryptographic Module

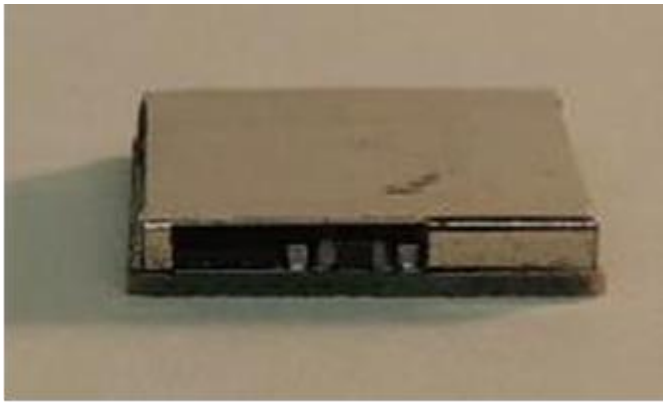

Figure 5: Back Side of the Cryptographic Module

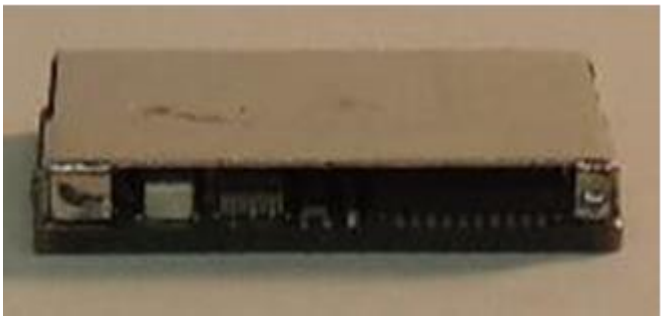

Figure 6: Left Side of the Cryptographic Module

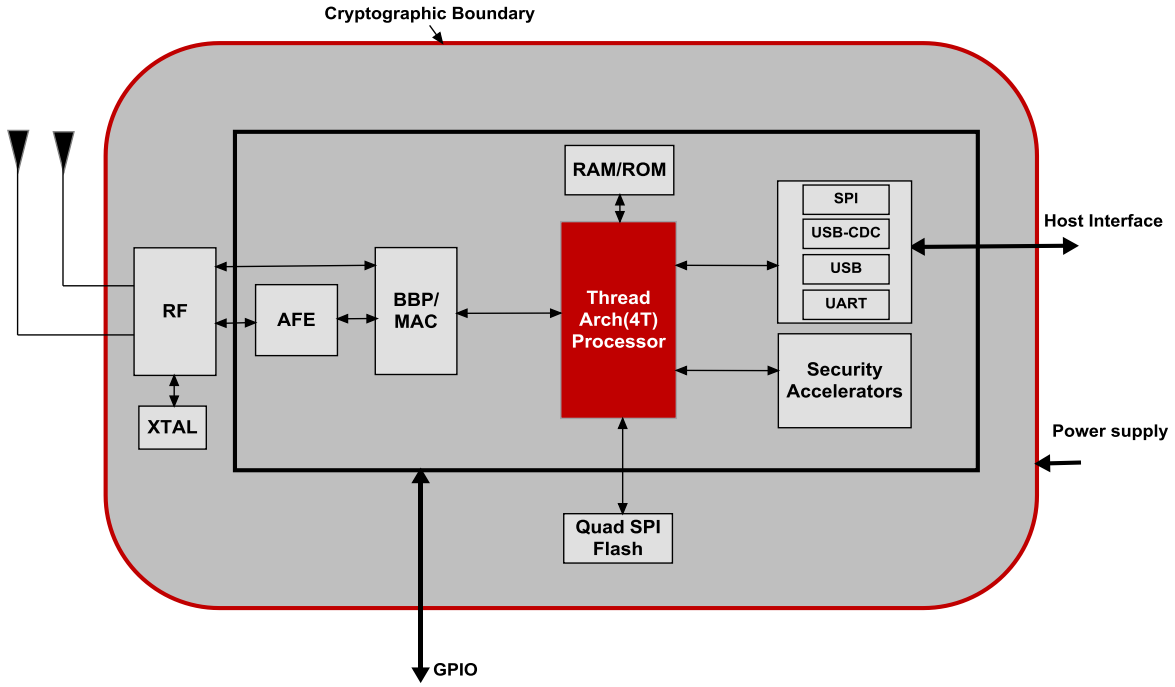

Figure 7: Cryptographic Module Block Diagram

Abbreviation	Description
RAM	Random Access Memory
ROM	Read Only Memory
SPI	Serial Peripheral Interface
USB-CDC	Universal Serial Bus- Communications Device Class
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transmitter
BBP	Base Band Processor
MAC	Media Access Control
AFE	Analog Front End
RF	Radio Frequency
XTAL	Crystal
GPIO	General - Purpose Input/Output

Table 1: Description of Block Diagram Abbreviations

3 Security Level Specification

Security Requirements Area	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	N/A
Cryptographic Key Management	1
EMI/EMC	1
Self-tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 2: Security Levels

4 Identification and Authentication Policy

The cryptographic module supports a Cryptographic Officer and User role. The roles are implicitly assumed by the operator.

The Cryptographic Officer can access all the services and CSPs of the module. Additionally, the Cryptographic Officer is responsible for initializing the module in a secure manner for the first time as described in Section 5.

The User can access all the services and CSPs of the module.

This module is designed to meet FIPS 140-2 Level 1 security requirements, therefore authentication requirements are not applicable.

The following table defines the roles, type of authentication, and associated authenticated data types supported by the cryptographic module:

Role	Type of Authentication	Authentication Data
Cryptographic Officer	N/A	N/A
User	N/A	N/A

Table 3: Roles and Required Authentication Mechanism

Authentication Mechanism	Strength of Mechanism: Random Attempted Breach	Strength of Mechanism: Multiple Consecutive Attempts in One-Minute Period
N/A	N/A	N/A

Table 4: Strengths of Authentication Mechanisms

5 Approved Modes of Operation

The cryptographic module supports a non-FIPS Approved mode of operation and a FIPS Approved mode of operation. CSPs defined in the FIPS Approved mode of operation cannot be accessed or shared while in the non-FIPS Approved mode of operation.

5.1 Non-FIPS Approved Mode

The cryptographic module may enter the non-FIPS Approved mode of operation during the initialization procedure by the Cryptographic Officer. The Cryptographic Officer must enter the fips_mode_enable command with the value “0” immediately after the init command during the initialization procedure:

fips_mode_enable: 0

The fips_mode_enable command is only available during the initialization procedure.

In order to enable the FIPS Approved mode from the non-FIPS Approved mode, the Cryptographic Officer must perform the Zeroize service and reboot the module.

NOTICE: If the module has been configured for FIPS Approved mode as described in Section 5.2 below, the Cryptographic Officer and User are required to abide by the restrictions documented in Section 8 below. In the event that the Cryptographic Officer or User violates or attempts to violate such restrictions, the module is in strict violation of this Security Policy and is deemed fully non-compliant and unfit for service to protect sensitive unclassified data with cryptography.

5.2 FIPS Approved Mode

FIPS approved mode can be enabled at power-up through the boot loader.

The Cryptographic Officer, physically present at the cryptographic boundary must follow the following guidelines to initialize the module in a secure manner for the first time:

1) Pins (TMS, TCK and TDI) shall be tied low.
2) Firmware Version RS9113.N00.WC.FIPS.OSI.1.8.2 with Bootloader version 1.7 / 1.8 is mandatory for using FIPS mode.
3) Power up the module.
4) In Binary mode enable FIPS mode by calling rsi_select_option() API with macro RSI_ENABLE_FIPS_MODE as an argument, which will write 0xAB46 value into HOST_INTERACT_REG_IN(0x41050034) register in the module.
5) Module will return 0xAB46 return code indicating the operator has successfully enabled the FIPS Approved Mode of Operation.
6) Disable boot loader interaction by calling rsi_select_option() API with macro RSI_ENABLE_BOOT_BYPASS as an argument, which will write 0xAB37 value into HOST_INTERACT_REG_IN(0x41050034) register.
7) Reboot the module.

The module will automatically perform power-up self-tests and upon successful completion, will enter FIPS Approved Mode of operation and sends card ready message to host.

Once the module successfully enters the FIPS Approved Mode of operation, the module cannot exit this mode unless the Cryptographic Officer explicitly performs the following procedures:

1) Perform the Zeroize service via rsi_fips_key_zeroization command.
2) Reboot the cryptographic module.
3) Enter the fips_mode_enable command with the value “0” immediately after the init command.

6 Approved and Non-Approved Algorithms

6.1 Approved Algorithms

CAVP Cert#	Algorithm	Standard	Model/ Method	Key Lengths, Curves, or Moduli	Use
3223	AES	FIPS-197, SP 800-38A	AES-CBC, AES-ECB,	128,192,256	Data encryption and decryption
2058	AES	FIPS-197, SP 800-38C	AES-CCM, AES-ECB	128	Data encryption and decryption
440	CVL	SP 800-135	TLS 1.0/1.1/1.2		Key derivation for TLS
C138	CVL	SP 800-56A	DHEphem	FB - L = 2048, N = 224; FC – L = 2048, N = 256	Key agreement and establishment
908	DRBG	SP 800-90A	SHA-256 HASH DRBG		Deterministic Random Bit generation
C138	DSA	FIPS 186-4		L = 2048 , N= 224 L = 2048, N= 256	Generation of private key and public key
2026	HMAC	FIPS 198-1	HMAC-SHA-1, HMAC-SHA-256		Message integrity and authentication
45	KDF	SP 800-108	HMAC-SHA-1		Key derivation
3223	KTS	SP 800-38F	AES-KW	128	key transportation
1639	RSA	FIPS 186-4	RSA PKCS1 V1.5	2048	Signature generation and verification
2661	SHS	FIPS 180-4	SHA-1, SHA-256		Message Digest

Table 5: Approved algorithms

6.2 Non-Approved, Allowed in FIPS Mode Algorithms

Algorithm	Caveat	Use
Hardware non-deterministic random number generator	Only used to seed the Hash_DRBG	For seeding Approved DRBG
Diffie-Hellman	Diffie-Hellman (CVL Cert. #C138 with CVL Cert. #440, key agreement; key establishment methodology provides 112 bits of encryption strength)	Key agreement and key establishment methodology
MD5	Used as per SP 800-135 Rev1 Section 4.2.1; MD5 is not exposed to the operator	Used in TLS v1.0 KDF
HMAC-MD5	Used as per SP 800-135 Rev1 Section 4.2.1; HMAC-MD5 is not exposed to the operator	Used in TLS v1.0 KDF
HMAC-MD5	Used to support RADIUS for operator authentication only; HMAC-MD5 is not exposed to the operator	RADIUS
RSA	RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength)	key wrapping , key establishment methodology

Table 6: Non-Approved, Allowed algorithms in FIPS mode

6.3 Non-Approved, Not Allowed in FIPS Mode Algorithms

The following algorithms are non-Approved and not allowed in the FIPS Approved Mode of Operation. These algorithms shall only be used in the non-FIPS Approved mode of operation with Non-Approved Services as per Table 10 in this security policy.

Algorithm	Use
AES-CCM (128-bit) (non-compliant)	Encryption and decryption
AES-CBC (128-bit, 256-bit) (non-compliant)	Encryption and decryption
DES	Encryption and decryption
Diffie-Hellman (512-bit, 1024-bit, 2048-bit, 4096-bit)	Key agreement and key establishment
DRBG (Hash_Drbg_256) (non-compliant)	Random Bit Generation
HMAC-MD4	Data authentication and integrity
HMAC-MD5 (non-compliant)	Data authentication and integrity
HMAC-SHA-1 (160-bit) (non-compliant)	Data authentication and integrity
HMAC-SHA-256 (256-bit) (non-compliant)	Data authentication and integrity
RC4	Encryption and decryption
RSA (512-bit, 1024-bit, 2048-bit, 4096-bit)	Key wrapping , key establishment, signature generation and signature verification
TLS v1.0 KDF (non-compliant)	Key derivation
TLS v1.2 KDF (non-compliant)	Key derivation

Table 7: Non-Approved, Not allowed algorithms in Fips mode

7 Physical Ports and Logical Interfaces

The cryptographic module does not contain a maintenance interface. The following table summarizes the physical ports and logical interfaces:

Physical Port	FIPS 140-2 Logical Interface
SDIO	(Not applicable; disabled; reserved for future use)
SPI Interface	Data Input/ Data Output/Control Input/ Status Output
USB 2.0	Data Input/ Data Output/Control Input/ Status Output
UART	Data Input/ Data Output/Control Input/ Status Output
RF	Data Input/ Data Output
I2S and PCM	(Not applicable; disabled; reserved for future use)
PMU Interface	Power
ULP PMU Interface	Power
GPIO	Data Input/ Data Output/Control Input/ Status Output
Power Port	Power
LED	Status Output; indicator that module is ON or OFF

Table 8: Specification of Cryptographic Module Physical Ports and Interfaces

8 Security Rules

The following specifies security rules under which the cryptographic module shall operate in accordance with FIPS 140-2:

- **NOTICE:** The Cryptographic Officer or User “shall not” enable the EAP-FAST service in FIPS Approved mode of operation. The type “FAST” shall not be used as argument to rsi_set_eap() API. If the operator enables the EAP-FAST service, it is an explicit violation of this Security Policy and the module is considered to be in non-FIPS mode.
- **NOTICE:** The Cryptographic Officer or User shall only load certificates with RSA 2048-bit keys into the module. RSA 1024/4096-bit certificates “shall not” be loaded into the module by the Cryptographic Officer or User; loading of aforementioned RSA 1024/4096-bit certificates is an explicit violation of this Security Policy and the module is considered to be in non-FIPS mode. In FIPS mode, loading RSA 1024 /4096-bit certificates will cause a failure of the TLS handshake, and module will report error 0x45
- **NOTICE:** The Cryptographic Officer or User shall only use Diffie-Hellman 2048-bit keys. Using Diffie-Hellman Keys other than 2048-bit is an explicit violation of this Security Policy and the module is considered to be in non-FIPS mode.
- **NOTICE:** The Cryptographic Officer or User “shall not” enable TCP/IP features DHCP(BIT(2)), HTTP(BIT(1)), HTTPS(BIT(12)), SNMP(BIT(9)), DNS(BIT(19)), FTP(BIT(15)), SNTP(BIT(16)), SMTP(BIT(20)), OTAF(BIT(30)), RAW-SOCKET(BIT(18)) and POP3(BIT(29)). These bits shall not be enabled in ‘rsi_opermode()’ API. Using these features is an explicit violation of this Security Policy and the module is considered to be in non-FIPS mode.The cryptographic module provides logical separation between all of the data input, control input, data output and status output interfaces. The module receives external power inputs through the defined power interface.
- The data output interface is inhibited during self-tests and when error states exist.
- When the cryptographic module is in an error state, it ceases to provide cryptographic services, inhibits all data outputs, and provides status of the error.
- The cryptographic module does not support multiple concurrent operators.
- The cryptographic module protects CSPs from unauthorized disclosure, unauthorized modification, and unauthorized substitution.
- The cryptographic module protects public keys from unauthorized modification, and unauthorized substitution.
- The cryptographic module satisfies the FCC EMI/EMC requirements for radios.

- The cryptographic module implements the following self-tests:

  1) Power-up tests

     * Cryptographic algorithm test
       - SHA-1 KAT
       - SHA-256 KAT
       - HMAC-SHA-1 KAT
       - HMAC-SHA-256 KAT
       - RSA 2048 SHA-256 Signature Generation KAT
       - RSA 2048 SHA-256 Signature Verification KAT
       - AES-128 CBC Encrypt KAT
       - AES-128 CBC Decrypt KAT
       - AES-256 CBC Encrypt KAT
       - AES-256 CBC Decrypt KAT
       - SP800-38F AES Key Wrap Encrypt KAT
       - SP800-38F AES Key Wrap Decrypt KAT
       - SP800-90A DRBG KAT
       - SP800-135 TLS v1.0 KDF KAT
       - SP800-135 TLS v1.2 KDF KAT
       - SP800-108 KDF KAT
       - AES-CCM KAT

     * Software/firmware test
       - Firmware integrity test (32-bit checksum)
       - Bootloader integrity test (32-bit checksum)

     * Critical functions test
       - SHA-1 checksum of configuration parameters

  2) Conditional tests

     * Firmware load test – RSA 2048 with SHA-256 Signature Verification

     * Manual key entry test - WPA2 Pre-shared Key (PSK) (256-bit) and RADIUS server password (1024-bit)

     * Continuous random number generator test
       - Continuous test on SP800-90A DRBG
       - Continuous test on non-Approved NDRNG

     * Bypass test: N/A

- The cryptographic module does not support bypass capability and does not implement bypass tests.
- The status indicator output by the module when power-on self-tests succeeds is indicated through a card ready message RSI_RSP_CARD_READY (0x89) with status 0x00 to the host.
- The status indicator output by the module when a power-on self-test fails is through FIPS Failure Indication message. with response frame type as RSI_RSP_FIPS_FAILURE (0xFA) to the host.
- The status indicator output by the module when a conditional self-test fails is through FIPS Failure Indication. with response frame type as RSI_RSP_FIPS_FAILURE (0xFA) to the host.
- The status indicator output by the module upon entry into the error state is through FIPS Failure Indication. with response frame type as RSI_RSP_FIPS_FAILURE (0xFA) to the host.
- Split-knowledge processes are not supported.
- All maintenance related services (i.e. maintenance role, physical maintenance interface, logical maintenance interface) are not applicable.
- Plaintext CSP output is not supported.
- The cryptographic module supports manual key entry and manual key entry test.
- The power interfaces cannot be used to drive power to external targets.
- The continuous comparison self-tests related to twin implementations are not applicable.
- The requirements of FIPS 140-2 Section 4.6 are not applicable; there exists no support for the execution of un-trusted code. All code loaded from outside the cryptographic boundary is cryptographically authenticated via the firmware load test.
- The requirements of FIPS 140-2 Section 4.11 are not applicable; the cryptographic module was not designed to mitigate specific attacks beyond the scope of FIPS 140-2.
- The module can clear the error condition only by performing hard reset followed by successful completion of self tests.
- On demand self tests can be performed by power cycling the module.
- The module will generate seeds for asymmetric key generation from the unmodified output of SP800-90A DRBG as per the requirements in SP800-133.

  Notes: The module does not generate keys for symmetric key algorithms, hence the ‘CKG’ entry is not applicable.

9 Access Control Policy

The access control policy lists the supported roles, services, cryptographic keys and CSPs, and types of access to the cryptographic keys and CSPs that are available to each of the authorized roles via the corresponding services.

9.1 Definition of Critical Security Parameters (CSPs)

The following are the CSPs contained in the module:

9.1.1 SP800-90A DRBG CSPs

- SP800-90A DRBG Seed Material
- SP800-90A DRBG Internal State

9.1.2 WPA2 802.11i CSPs and Keys

- SP800-90A DRBG CSPs (NOTE: Refer to Section 9.1.1 above)
- WPA2 Pre-shared Key (PSK) (256-bit)
- 802.11i KDF Internal State
- 802.11i Temporal keys (AES-CCM 128-bit)
- 802.11i MIC keys (KCK) (HMAC-SHA-1 128-bit)
- 802.11i Key Encryption Key (KEK) (AES-KW 128-bit)
- 802.11i Group Temporal Keys (GTK) (AES-CCM 128-bit)

9.1.3 TLS CSPs and Keys

- SP800-90A DRBG CSPs (NOTE: Refer to Section 9.1.1 above)
- Diffie-Hellman Private keys (2048-bit)
- Diffie-Hellman Shared secret (2048-bit)
- TLS KDF Internal State
- TLS Encryption Key (AES-CBC 128-bit, 256-bit)
- TLS Integrity Key (HMAC-SHA-1 160-bit, HMAC-SHA-2 256-bit)
- TLS Master Secret (384-bit)
- TLS Pre-Master Secret (384-bit)
- RSA Private Key of client certificate (RSA 2048-bit)

9.1.4 EAP-TLS CSPs and Keys

- TLS CSPs and Keys
- EAP-TLS Master Session Key (MSK) (512-bit)

9.1.5 EAP-TTLS CSPs and Keys

- EAP-TLS CSPs and Keys (NOTE: Refer to Section 9.1.4 above)
- EAP-TTLS Master Session Key (MSK) (512-bit)
- RADIUS server password (1024-bit)

9.1.6 EAP-PEAP CSPs and Keys

- EAP-TLS CSPs and Keys (NOTE: Refer to Section 9.1.4 above)
- EAP-PEAP Master Session Key (MSK) (512-bit)
- RADIUS server password (1024-bit)

9.1.7 Wireless Firmware Update CSPs

- WPA2 802.11i CSPs and Keys (NOTE: Refer to Section 9.1.2 above)

9.2 Definition of Public Keys

The following are the public keys contained in the module:

9.2.1 TLS Public Keys

- Module Diffie-Hellman Public Keys (2048-bit)
- Server Diffie-Hellman Public Keys (2048-bit)
- CA certificate RSA Public Key (RSA 2048-bit)
- Intermediate CA certificate RSA Public Key (RSA 2048-bit)

9.2.2 EAP-TLS Public Keys

- TLS Public Keys (NOTE: Refer to section 9.2.1 above)
- Module certificate EAP-TLS (RSA 2048-bit)
- Server certificate EAP-TLS (RSA 2048-bit)

9.2.3 EAP-TTLS Public Keys

- TLS Public Keys (NOTE: Refer to section 9.2.1 above)
- Server certificate EAP-TTLS (RSA 2048-bit)

9.2.4 EAP-PEAP Public Keys

- TLS Public Keys (NOTE: Refer to section 9.2.1 above)
- Server certificate EAP-PEAP (RSA 2048-bit)

9.2.5 Firmware Update Public Key

- Firmware Update Public Key

User Role	Cryptographic Officer Role	Service	Type(s) of Access to Cryptographic Keys, CSPs and Public Keys W = Write the item into memory Z = Zeroize U = Use
	X	Module Initialization	N/A
X	X	Self-Tests	N/A
X	X	Show-status	N/A
X	X	Zeroize	Z – All CSPs
X	X	Manual key entry	W – WPA2 Pre-shared Key (PSK) (256-bit) W – RADIUS server password (1024-bit)
X	X	Import Certificates	W – RSA Private Key of client certificate (RSA 2048-bit) W – Module certificate EAP-TLS (RSA 2048-bit) W – CA certificate RSA Public Key (RSA 2048-bit)
X	X	Firmware Update	U – Firmware Update CSPs U – Firmware Update Public Key
X	X	TLS	U - TLS CSPs and Keys U - TLS Public keys
X	X	EAP-TLS	U – EAP-TLS CSPs and Keys U – EAP-TLS Public Keys
X	X	EAP-TTLS	U – EAP-TTLS CSPs and Keys U – EAP-TTLS Public Keys
X	X	EAP-PEAP	U – EAP-PEAP CSPs and Keys U – EAP-PEAP Public Keys
X	X	WPA2 802.11i	U – WPA2 802.11i CSPs and Keys U – EAP-TLS Master Session Key (MSK) (512-bit) U – EAP-TTLS Master Session Key (MSK) (512-bit) U – EAP-PEAP Master Session Key (MSK) (512-bit)

Table 9: Services Authorized for Roles, Access Rights within Services

Certain services are available within the Non-FIPS Approved mode of operation, which are otherwise not available in the FIPS Approved Mode of operation as described in Table 10 below:

User Role	Cryptographic Officer Role	Service	Non-FIPS Approved Algorithms	Description
X	X	TLS	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (1024-bit), Diffie-Hellman (512-bit, 1024-bit)	Provide Wireless connection with Access Point
x	x	EAP-TLS	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (1024-bit), Diffie-Hellman (512-bit, 1024-bit)	
X	X	EAP-TTLS	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (1024-bit), Diffie-Hellman (512-bit, 1024-bit)	
X	X	EAP-PEAP	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (1024-bit), Diffie-Hellman (512-bit, 1024-bit)	
X	X	Open	No Ciphers are used (no cryptography)	

User Role	Cryptographic Officer Role	Service	Non-FIPS Approved Algorithms	Description
X	X	WEP	RC4	
X	X	WPA	RC4	
X	X	EAP-FAST	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (1024-bit), Diffie-Hellman (512-bit, 1024-bit)	
X	X	DHCP	No Ciphers are used (no cryptography)	
X	X	HTTP	No Ciphers are used (no cryptography)	
X	X	HTTPS	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (512-bit, 1024-bit, 2048-bit, 4096-bit), Diffie-Hellman (512-bit, 1024-bit, 2048-bit, 4096-bit), AES-CCM (128-bit) (non-compliant), AES-CBC (128-bit, 256-bit) (non-compliant), HMAC-SHA-1 (160-bit) (non-compliant), HMAC-SHA-256 (256-bit) (non-compliant), TLS v1.0 KDF (non-compliant), TLS v1.2 KDF (non-compliant), DRBG (Hash_Drbg_256) (non-compliant)	

User Role	Cryptographic Officer Role	Service	Non-FIPS Approved Algorithms	Description
X	X	SNMP	SNMPv2 - No Ciphers are used (no cryptography)	
X	X	DNS	No Ciphers are used (no cryptography)	
X	X	FTP	No Ciphers are used (no cryptography)	
X	X	SNTP	No Ciphers are used (no cryptography)	
X	X	SMTP	RC4, DES, HMAC-MD5 (non-compliant), HMAC-MD4, RSA (512-bit, 1024-bit, 2048-bit, 4096-bit), Diffie-Hellman (512-bit, 1024-bit, 2048-bit, 4096-bit), AES-CCM (128-bit) (non-compliant), AES-CBC (128-bit, 256-bit) (non-compliant), HMAC-SHA-1 (160-bit) (non-compliant), HMAC-SHA-256 (256-bit) (non-compliant), TLS v1.0 KDF (non-compliant), TLS v1.2 KDF (non-compliant), DRBG (Hash_Drbg_256) (non-compliant)	
X	X	OTAF	No Ciphers are used (no cryptography)	
X	X	RAW-SOCKET	No Ciphers are used (no cryptography)	
X	X	POP3	No Ciphers are used (no cryptography)	

Table 10: Non-Approved Services only allowed in Non-FIPS Approved Mode

10 Physical Security Policy

The cryptographic module implements the following physical security mechanisms:

- Production grade components.

The following table summarizes the actions required by the Cryptographic Officer Role to ensure that physical security is maintained.

Physical Security Mechanisms	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Production grade components	N/A	N/A

Table 11: Inspection/Testing of Physical Security Mechanisms

11 Mitigation of Other Attacks Policy

The cryptographic module is not designed to mitigate against attacks outside the scope of FIPS 140-2.

Other Attacks	Mitigation Mechanism	Specific Limitations
N/A	N/A	N/A

Table 12: Mitigation of Other Attacks

12 References

- FIPS PUB 140-2
- FIPS PUB 140-2 DTR
- FIPS PUB 140-2 Implementation Guidance
- FIPS 180-4
- FIPS 186-4
- FIPS 197
- FIPS 198
- NIST SP800-38C
- NIST SP800-38F
- NIST SP800-90A
- NIST SP800-108
- NIST SP800-131A
- NIST SP800-133
- NIST SP800-135
- RFC 3748
- RFC 5247
- RFC 5281
- NIST SP800-52

13 Appendix A: Critical Security Parameters

The module supports the following critical security parameters:

1. SP800-90A DRBG Seed Material:
   Description - The seed (384-bit) for the Approved DRBG
   Generation - Internally from hardware non-deterministic random number generator; acceptable as per FIPS 140-2 Section 4.7.1
   Establishment – N/A
   Storage - Plaintext; stored in Hardware Register and RAM
   Entry - N/A
   Output - N/A
   Key-To-Entity – DRBG Process
   Destruction - Actively overwritten via Zeroize service

2. SP800-90A DRBG Internal State:
   Description - The internal state of the DRBG (Note: The values of V (440-bit) and C (440-bit) are the “secret values” of the internal state)
   Generation - Approved SP800-90A HASH_DRBG; Approved as per FIPS 140-2 Annex C
   Establishment – N/A
   Storage - Plaintext; stored in RAM
   Entry - N/A
   Output - N/A
   Key-To-Entity – DRBG Process
   Destruction - Actively overwritten via Zeroize service

3. WPA2 Pre-shared key (PSK) (256-bit):
   Description - 256-bit shared secret used for pre-shared key authentication and session key establishment; used as PMK for 802.11i KDF
   Generation - N/A
   Establishment – N/A
   Storage - Plaintext; stored in Flash and RAM
   Entry - Manually transported, electronically entered in plaintext through command from host (Manual Key Entry Test)
   Output - N/A
   Key-To-Entity - WPA2 protocol
   Destruction - Actively overwritten via Zeroize service

4. 802.11i KDF Internal State:
   Description - Used for key derivation (SP800-108 KDF(HMAC-SHA-1) with 256-bit PMK as input) to calculate the WPA2 session keys
   Generation – SP800-108 KDF (HMAC-SHA-1); acceptable as per FIPS 140-2 IG 7.10
   Establishment – N/A
   Storage - Plaintext; stored in RAM
   Entry - N/A
   Output - N/A
   Key-To-Entity - WPA2 protocol
   Destruction - Actively overwritten via Zeroize service

5. 802.11i Temporal keys (AES-CCM 128-bit):

Description - AES-CCM 128-bit keys used for session encryption/decryption
Generation – N/A
Establishment – Dynamically via SP800-108 KDF(HMAC-SHA-1); acceptable as per FIPS 140-2 IG 7.10
Storage - Plaintext; stored in Hardware Register and RAM
Entry - N/A
Output - N/A
Key-To-Entity - WPA2 protocol
Destruction - Actively overwritten via Zeroize service

6. 802.11i MIC keys (KCK) (HMAC-SHA-1 128-bit):
Description – Key Confirmation Keys (HMAC-SHA-1 128-bit) used for message authentication during session establishment
Generation – N/A
Establishment – Dynamically via SP800-108 KDF (HMAC-SHA-1); acceptable as per FIPS 140-2 IG 7.10
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity - WPA2 protocol
Destruction - Actively overwritten via Zeroize service

7. 802.11i Key Encryption Key (KEK) (AES-KW 128-bit):
Description – AES 128-bit key used for key wrapping of the 802.11i Group Temporal Key (GTK)
Generation – N/A
Establishment – Dynamically via SP800-108 KDF (HMAC-SHA-1); acceptable as per FIPS 140-2 IG 7.10
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity - WPA2 protocol
Destruction - Actively overwritten via Zeroize service

8. 802.11i Group Temporal Keys (GTK) (AES-CCM 128-bit):
Description - 802.11i session key (AES-CCM 128-bit) for broadcast communications
Generation - N/A
Establishment – Key Transport: AES key wrapped with 802.11i Key Encryption Key (KEK) (AES-KW 128-bit); Approved as per FIPS 140-2 IG D.9
Storage - Plaintext; stored in Hardware Register and RAM
Entry – Key Transport: AES key wrapped with 802.11i Key Encryption Key (KEK) (AES-KW 128-bit); Approved as per FIPS 140-2 IG D.9
Output - N/A
Key-To-Entity - WPA2 protocol
Destruction - Actively overwritten via Zeroize service

9. Diffie-Hellman Private Keys (2048-bit)

Description – Used in TLS, EAP-TLS, EAP-TTLS and EAP-PEAP to establish a shared secret
Generation – As per SP800-133 Section 6.2, the random value (K) needed to generate key pairs for the finite field is the output of the SP800-90A DRBG; this is an allowed method as per FIPS 140-2 IG D.8 Scenario 4
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity –TLS, EAP-TLS, EAP-TTLS, and EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

10. Diffie-Hellman shared secret (2048-bit)
Description – 2048-bit Shared Secret used to facilitate TLS, EAP-TLS, EAP-TTLS, and EAP-PEAP.
This is also known as a TLS Pre-Master Secret.
Generation – N/A
Establishment- As per SP800-56A section 6.1.2.1
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity –TLS, EAP-TLS, EAP-TTLS, and EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

11. TLS KDF Internal State
Description – Values of the TLS v1.0 and TLS v1.2 KDF internal state used in TLS, EAP-TLS, EAP-TTLS, and EAP-PEAP. For TLS v1.0 KDF, module implements MD5 and SHA-1. For TLS v1.2 KDF module implements SHA-256 PRF as per SP800-135.
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity – TLS,EAP-TLS, EAP-TTLS, and EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

12. TLS Encryption Key (AES-CBC 128-bit, 256-bit):
Description - AES-CBC (128, 256 bit) key used to encrypt TLS, EAP-TLS, EAP-TTLS and EAP-PEAP session data
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

13. TLS Integrity Key (HMAC-SHA-1 160-bit, HMAC-SHA-2 256 bit):

Description – HMAC keys used for TLS, EAP-TLS, EAP-TTLS and EAP-PEAP integrity protection
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

14. TLS Master Secret (384-bit):
Description – TLS, EAP-TLS, EAP-TTLS, EAP-PEAP shared secret (Master Secret)
Generation – N/A
Establishment – Master secret is established with TLS Pre-Master Secret or Diffie-Hellman shared secret as an input to TLS v1.0/TLS v1.2 KDF as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry – N/A
Output - N/A
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS, EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

15. TLS Pre-Master Secret (384-bit):
Description – TLS, EAP-TLS, EAP-TTLS, EAP-PEAP shared secret (Pre-Master Secret)
Generation – When the module behaves as a TLS client, shared secret is generated by the Approved SP800-90A DRBG; Approved as per FIPS 140-2 Annex C
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry – When the module behaves as a TLS Client, this entry is N/A. When the module behaves as a TLS Server, the shared secret is entered RSA key wrapped by the module; allowed as per FIPS 140-2 IG D.9
Output – When the module behaves as a TLS Server, this entry is N/A. When the module behaves as a TLS Client, the shared secret is output RSA key wrapped with the Server's Public Key.
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS, EAP-PEAP protocols
Destruction - Actively overwritten via Zeroize service

16. EAP-TLS Master Session Key (MSK) (512-bit):
Description – 512-bit session key, where 256-bit MSB are used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF with TLS Master Secret as input as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity - EAP-TLS protocol
Destruction - Actively overwritten via Zeroize service

17. RSA Private Key of client certificate (RSA 2048-bit):

Description - RSA 2048-bit private key portion of the Module certificate EAP-TLS for Digital Signature generation within EAP-TLS
Generation – N/A
Establishment – N/A
Storage - Plaintext; stored in Flash and RAM
Entry - Manually transported, electronically entered in plaintext as per FIPS 140-2 IG 7.7 through command from host
Output - N/A
Key-To-Entity – EAP-TLS Protocol
Destruction - Actively overwritten via Zeroize service

18. RADIUS server password (1024-bit):
Description - Password used to authenticate the RADIUS server during EAP-TTLS and EAP-PEAP protocols
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in Flash and RAM
Entry - Manually transported, electronically entered in plaintext through command from host (Manual Key Entry Test)
Output - N/A
Key-To-Entity - RADIUS server
Destruction - Actively overwritten via Zeroize service

19. EAP-TTLS Master Session Key (MSK) (512-bit):
Description - 512-bit session key, where 256-bit MSB are used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF with TLS Master Secret as input as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity - EAP-TTLS protocol
Destruction - Actively overwritten via Zeroize service

20. EAP-PEAP Master Session Key (MSK) (512-bit):
Description - 512-bit session key, where 256-bit MSB are used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0/TLS v1.2 KDF with TLS Master Secret as input as per SP800-135 Section 4.2.1 & 4.2.2; allowed method as per FIPS 140-2 IG D.8 Scenario 4
Storage - Plaintext; stored in RAM
Entry - N/A
Output - N/A
Key-To-Entity - EAP-PEAP protocol
Destruction - Actively overwritten via Zeroize service

14 Appendix B: Public Keys

The module supports the following public keys:

1. Module Diffie-Hellman Public Keys (2048-bit)
   Description – Used in TLS, EAP-TLS, EAP-TTLS and EAP-PEAP to establish a shared secret
   Generation – As per FIPS 186-4 where public key is derived from the private key. SP800-133 Section 6.2, the random value (K) needed to generate key pairs for the finite field is the output of the SP800-90A DRBG; this is an allowed method as per FIPS 140-2 IG D.8 Scenario 4
   Establishment – N/A
   Storage - Plaintext; stored in RAM
   Entry - N/A
   Output – Plaintext; output to Server during TLS exchanges
   Key-To-Entity - TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols

2. Server Diffie-Hellman Public Keys (2048-bit)
   Description – Used in TLS, EAP-TLS, EAP-TTLS, and EAP-PEAP to establish a shared secret
   Generation – N/A
   Establishment – N/A
   Storage - Plaintext; stored in RAM
   Entry – Plaintext; entered during TLS exchanges
   Output - N/A
   Key-To-Entity - TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols

3. Module certificate EAP-TLS (RSA 2048-bit)
   Description - Client certificate in EAP-TLS used by the server to authenticate the client
   Generation - N/A
   Establishment – N/A
   Storage - Plaintext; stored in Flash and RAM
   Entry - Manually transported, electronically entered in plaintext through command from host Output - Plaintext; output to EAP-TLS server during EAP-TLS hand shake
   Key-To-Entity – EAP-TLS protocol

4. Server certificate EAP-TLS (RSA 2048-bit):
   Description - Server certificate in EAP-TLS used by the client to authenticate the server
   Generation - N/A
   Establishment – N/A
   Storage - Plaintext; stored in RAM
   Entry – Plaintext; entered into the module during EAP-TLS exchanges
   Output – N/A
   Key-To-Entity – EAP-TLS protocol

5. Server certificate EAP-TTLS (RSA 2048-bit):
   Description - Server certificate in EAP-TTLS used by the client to authenticate the server

Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry – Plaintext; entered into the module during EAP-TTLS exchanges
Output – N/A
Key-To-Entity – EAP-TTLS protocol

6. Server certificate EAP-PEAP (RSA 2048-bit):
Description - Server certificate in EAP-PEAP used by the client to authenticate the server
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry – Plaintext; entered into the module during EAP-PEAP exchanges
Output – N/A
Key-To-Entity – EAP-PEAP protocol

7. CA certificate RSA Public Key (RSA 2048-bit):
Description - Public key (RSA 2048-bit) in the CA certificate
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in Flash and RAM
Entry - Manually transported, electronically entered in plaintext through command from host
Output - N/A
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols

8. Intermediate CA certificate RSA Public Key (RSA 2048-bit):
Description - Public key (RSA 2048-bit) in the CA certificate
Generation – N/A
Establishment – N/A
Storage - Plaintext; stored in Flash and RAM
Entry – Plaintext; when behaving as TLS client, entered into the module during TLS, EAP-TLS, EAP-TTLS, EAP-PEAP exchanges. When behaving as TLS server, it is manually transported, electronically entered in plaintext through command from host
Output - When behaving as TLS server, output in plain text during the TLS, EAP-TLS, EAP-TTLS, EAP-PEAP exchanges
Key-To-Entity – TLS, EAP-TLS, EAP-TTLS and EAP-PEAP protocols

9. Firmware Update Public Key
Description - RSA 2048-bit key used to verify RSA 2048 with SHA-256 signatures for secure Firmware Update Service
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in Flash and RAM
Entry – N/A; installed into the module in the secure factory
Output - N/A
Key-To-Entity – Firmware Update process

*****

15 Revision History

Revision	Author	Description	Date: (DD.MM.YYYY)
04	Bill Dowd	Initial Release	25.11.2015
05	Bill Dowd	Updated for New Redpine FW 1.8.2 with bootloaders 1.7 / 1.8	30.08.2019
06	Bill Dowd	Updated CVL and DSA cert numbers in Approved Algorithms table	30.09.2020