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

Table of Contents

1 Introduction

The DRAEGER WCM9113 802.11ABGN VG2 (HW Revision: MS32018 Rev. 02; FW Version: VG2 with Bootloader version 1.7) herein after referred to as “cryptographic module” or “module”, is a FIPS 140-2 Level 1 multi-chip embedded cryptographic module that has been licensed to Draeger for exclusive use by Redpine Signals.

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

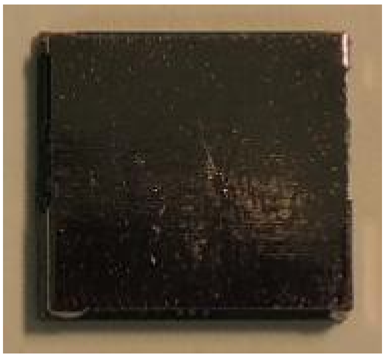

Figure 2 Bottom side of the Cryptographic Module

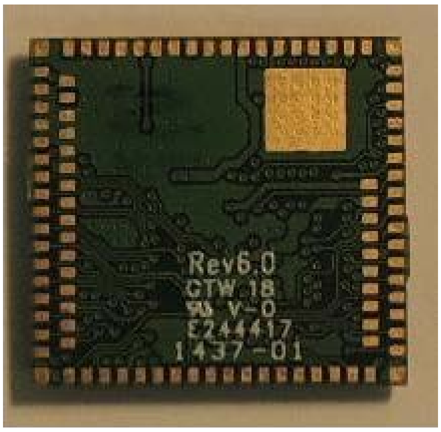

Figure 3 Front side of the Cryptographic Module

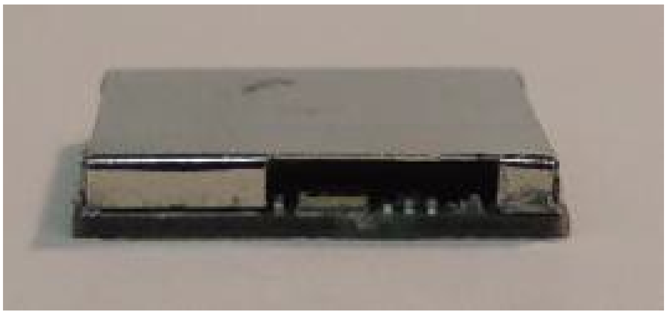

Figure 4 Right side of the Cryptographic Module

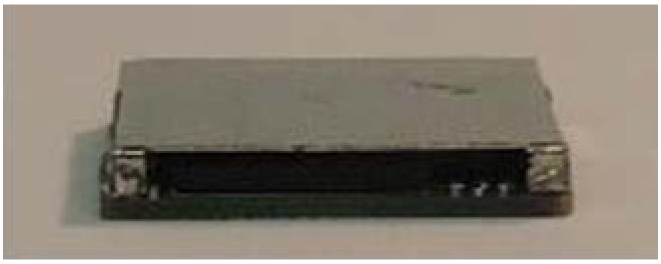

Figure 5 Back side of the Cryptographic Module

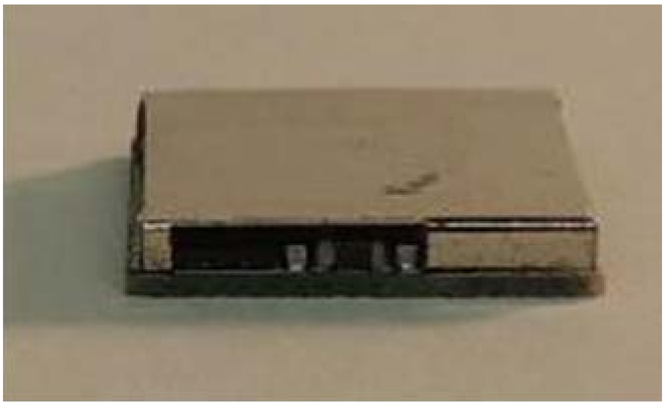

Figure 6 Left side of the Cryptographic Module

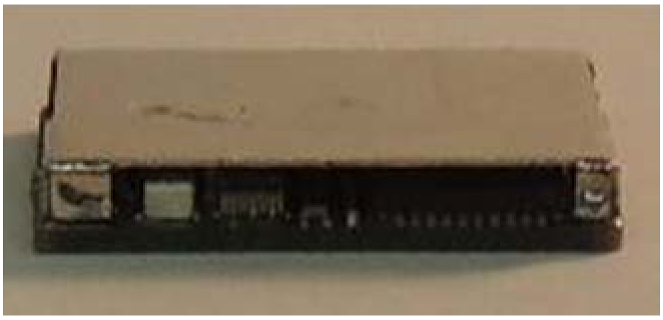

Figure 7 Cryptographic Module Block Diagram

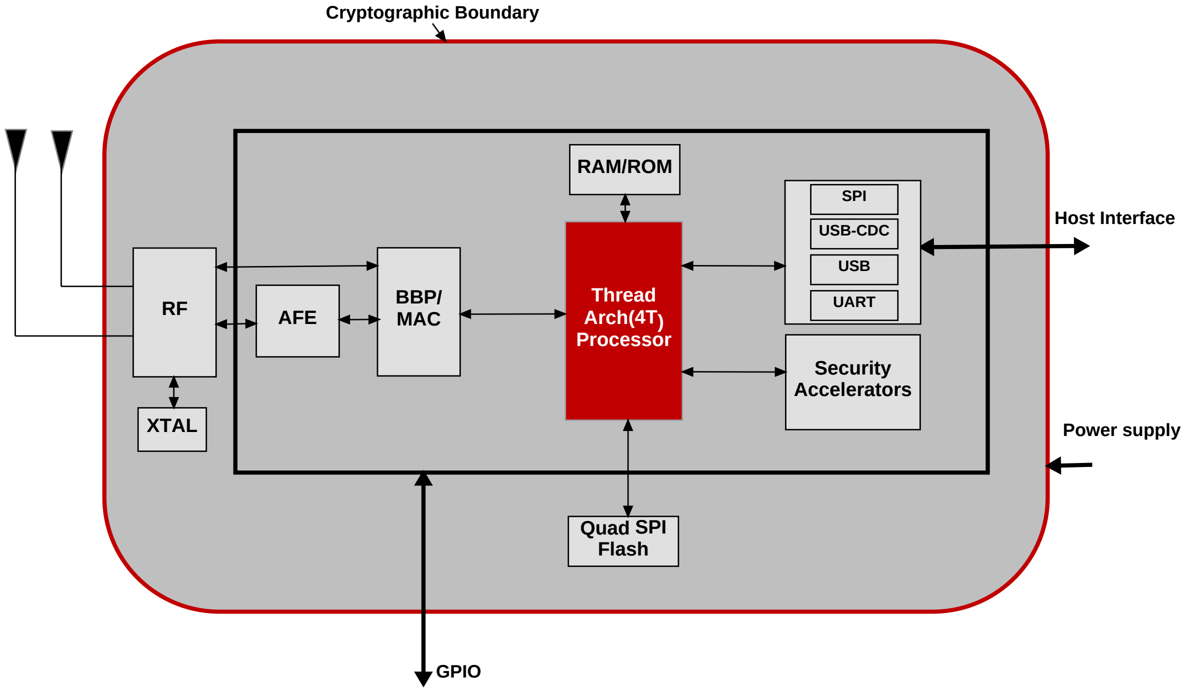

Table 1 Description of Block Diagram Abbreviations

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

3 Security Level Specification

Table 2 Security Levels

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

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:

Table 3 Roles and Required Authentication Mechanism

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

Table 4 Strengths of Authentication Mechanisms

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

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 VG2 with Bootloader version 1.7 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 it will enter the FIPS Approved Mode of operation.

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

The cryptographic module supports the following Approved algorithms:

- AES with 128-bit key and 256-bit key in CBC mode Encrypt/Decrypt (Cert. #3223) *[NOTE: The underlying ECB mode was validated for Assurance Purposes although it is not exposed directly within any callable service; AES-192 on this algorithm validation certificate is latent functionality and is not exposed directly within any callable service]*
  - Effective Strength: 112-bits
- KTS (AES Cert. #3223; key establishment methodology provides 112 bits of encryption strength)
- AES-128 CCM (Cert. #2058) *[NOTE: The underlying ECB mode was validated for Assurance Purposes although it is not exposed directly within any callable service]*
  - Effective Strength: 112-bits
- SHA-1, SHA-256 (Cert. #2661)
  - Effective Strength: 112-bits
- HMAC-SHA-1 (Cert. #2026) *[NOTE: HMAC-SHA-256 on this algorithm validation certificate is latent functionality and is not exposed directly within any callable service]*
  - Effective Strength: 112-bits
- RSA PKCS1 V1.5 with 2048-bit key and SHA-256 for Digital Signature Generation/Verification (Cert. #1639)
  - Effective Strength: 112-bits
- SP800-90A DRBG SHA-256 HASH_DRBG (Cert. #908)
  - Effective Strength: 112-bits
- SP800-108 KDF CTR_Mode with HMAC-SHA-1 (Cert. #45)
  - Effective Strength: 112-bits
- CVL: SP800-135 TLS v1.0 KDF with HMAC-SHA-1 (Cert. #440) *[NOTE: The TLS v1.2 KDF on this algorithm validation certificate “cannot” be used in FIPS Approved Mode or the non-FIPS Approved mode; TLS v1.2 is latent functionality]*
  - Effective Strength: 112-bits

The following protocols have not been reviewed or tested by the CAVP and CMVP:

- TLS v1.0
- TLS v1.2 (TLS v1.2 is latent functionality)

6.2 Non-Approved, Allowed in FIPS mode algorithms

The cryptographic module supports the following non-Approved algorithms:

- Hardware non-deterministic random number generator (for seeding Approved DRBG)
- Diffie-Hellman (key agreement; key establishment methodology provides 112 bits of encryption strength)
- RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength)
- HMAC-MD5 (Used in TLS v1.0 KDF only)
- HMAC-MD5 (Used in RADIUS only)

6.3 Non-Approved, not Allowed in FIPS mode algorithms

The cryptographic module supports the following Non-Approved algorithms only in the Non-FIPS Approved Mode of operation:

- RC4
- DES
- HMAC-MD4
- HMAC-MD5
- RSA-1024 (non-compliant)
- Diffie-Hellman (512-bit, 1024-bit)

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:

Table 5 Specification of cryptographic module physical ports and interfaces

Physical Port	FIPS 140-2 Logical Interface
SDIO	Data Input/ Data Output/Control Input/ Status Output
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.

8 Security rules

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

- **NOTICE:** For the avoidance of doubt, it is hereby re-iterated that the TLS v1.2 “is not” available in FIPS Mode and non-FIPS mode; TLS v1.2 is latent functionality.
- **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-bit certificates “shall not” be loaded into the module by the Cryptographic Officer or User; loading of aforementioned RSA 1024-bit certificates 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 use Diffie-Hellman 2048-bit keys. Using Diffie-Hellman Public Keys other than 2048-bit 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 *[NOTE: HMAC-SHA-256 is latent functionality and is not exposed directly within any callable service]*
- 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
- SP800-108 KDF KAT
- AES-CCM KAT

** Software/firmware test*

- Firmware integrity test (32-bit checksum)
- Boot-loader 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)
- ➢ Pairwise Consistency Test - N/A
- ➢ 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 to the host.
- The status indicator output by the module when a power-on self-test fails is through FIPS Failure Indication message.

- The status indicator output by the module when a conditional self-test fails is through FIPS Failure Indication message.
- The status indicator output by the module upon entry into the error state is through FIPS Failure Indication message.
- 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 against specific attacks beyond the scope of FIPS 140-2.

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 list displays the CSPs contained in the module:

9.1.1 SP800-90A DRBG CSPs

- SP800-90A DRBG Seed Material
- SP800-90A DRBG Entropy Input String
- 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-bits)
- 802.11i MIC keys (KCK) (HMAC-SHA-1 128-bits)
- 802.11i Key Encryption Key (KEK) (AES 128-bit)
- 802.11i Group Temporal Key (GTK) (AES-CCM 128-bit)

9.1.3 EAP-TLS CSPs and Keys

- SP800-90A DRBG CSPs (NOTE: Refer to Section 9.1.1 above)
- Diffie-Hellman Private keys (2048-bit)
- TLS v1.0 KDF Internal State
- EAP-TLS Encryption Key (AES-CBC 128-bit, 256-bit)
- EAP-TLS Integrity Key (HMAC-SHA-1 160-bit)
- EAP-TLS Master Secret
- EAP-TLS Pre-Master Secret
- EAP-TLS Master Session Key (MSK)

- RSA Private Key of client certificate (RSA 2048-bit)

9.1.4 EAP-TTLS CSPs and Keys

- EAP-TLS CPSs and Keys (NOTE: Refer to Section 9.1.3 above)
- EAP-TTLS Master Session Key (MSK) (256-bit)
- RADIUS server password (1024-bit)

9.1.5 EAP-PEAP CSPs and Keys

- EAP-TLS CSPs and Keys (NOTE: Refer to Section 9.1.3 above)
- EAP-PEAP Master Session Key (MSK) (256-bit)
- RADIUS server password (1024-bit)

9.1.6 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 EAP-TLS Public Keys

- Client Diffie-Hellman Public Keys (2048-bit)
- Server Diffie-Hellman Public Keys (2048-bit)
- Client EAP-TLS RSA Public Key used in the client certificate (RSA 2048-bit)
- Server EAP-TLS RSA Public Key used in the server certificate (RSA 2048-bit)
- CA Certificate RSA Public Key (RSA 2048-bit)

9.2.2 EAP-TTLS Public Keys

- Client Diffie-Hellman Public Keys (2048-bit)
- Server Diffie-Hellman Public Keys (2048-bit)
- Server EAP-TTLS RSA Public Key (RSA 2048-bit)
- CA Certificate RSA Public Key (RSA 2048-bit)

9.2.3 EAP-PEAP Public Keys

- Client Diffie-Hellman Public Keys (2048-bit)
- Server Diffie-Hellman Public Keys (2048-bit)
- Server EAP-PEAP RSA Public Key (RSA 2048-bit)
- CA Certificate RSA Public Key (RSA 2048-bit)

9.2.4 Firmware Update Public Key

- Firmware Update Public Key

Table 6 Services Authorized for Roles, Access Rights within Services

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 – Client EAP-TLS RSA Public Key used in the client certificate (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	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) (256-bit) U – EAP-TTLS Master Session Key (MSK) (256-bit) U – EAP-PEAP Master Session Key (MSK) (256-bit)

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 7 below:

Table 7 Non-Approved Services only allowed in Non-FIPS Approved Mode

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

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.

Table 8 Inspection/Testing of Physical Security Mechanisms

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

11 Mitigation of Other Attacks Policy

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

Table 9 Mitigation of Other Attacks

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

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

13 Appendix A: Critical Security Parameters

The module supports the following critical security parameters:

1. SP800-90A DRBG Seed Material:
   Description - The seed 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
   Entry - N/A
   Output - N/A
   Key-To-Entity - Process
   Destruction - Actively overwritten via Zeroize service

2. SP800-90A DRBG Entropy Input String:
   Description – The entropy input string 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
   Entry - N/A
   Output - N/A
   Key-To-Entity - Process
   Destruction - Actively overwritten via Zeroize service

3. SP800-90A DRBG Internal State:
   Description - The internal state of the DRBG (Note: The values of V and Key are the “secret values” of the internal state)
   Generation - Approved SP800-90A 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 - Process
   Destruction - Actively overwritten via Zeroize service

4. 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
   Output - N/A
   Key-To-Entity - WPA2 protocol
   Destruction - Actively overwritten via Zeroize service

5. 802.11i KDF Internal State:
   Description - Used for key derivation (SP800-108 KDF with 256-bit PMK as input) to calculate the WPA2 session keys

Generation – N/A
Establishment – Dynamically via SP800-108 KDF; 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

6. 802.11i Temporal keys (AES-CCM 128-bits):
Description - AES-CCM 128-bit keys used for session encryption/decryption
Generation – N/A
Establishment – Dynamically via SP800-108 KDF; 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 MIC keys (KCK) (HMAC-SHA-1 128-bits):
Description – Key Confirmation Keys (HMAC-SHA-1 128-bits) used for message authentication during session establishment
Generation – N/A
Establishment – Dynamically via SP800-108 KDF; 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 Key Encryption Key (KEK) (AES 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; 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

9. 802.11i Group Temporal Key (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 128-bit); Approved as per FIPS 140-2 IG D.9
Storage - Plaintext; stored in RAM
Entry – Key Transport: AES key wrapped with 802.11i Key Encryption Key (KEK) (AES 128-bit)
Output - N/A
Key-To-Entity - WPA2 protocol
Destruction - Actively overwritten via Zeroize service

10. Diffie-Hellman Private Keys (2048-bits)

Description – Used in 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 - EAP-TLS, EAP-TTLS, and EAP-PEAP protocols

Destruction - Actively overwritten via Zeroize service

11. TLS v1.0 KDF Internal State

Description – Values of the TLS v1.0 KDF internal state used in EAP-TLS, EAP-TTLS, and EAP-PEAP

Generation – N/A

Establishment – TLS v1.0 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 - EAP-TLS, EAP-TTLS, and EAP-PEAP protocols

Destruction - Actively overwritten via Zeroize service

12. EAP-TLS Encryption Key (AES-CBC 128-bit, 256-bit):

Description - AES-CBC (128, 256 bit) key used to encrypt EAP-TLS session data

Generation – N/A

Establishment – TLS v1.0 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 - EAP-TLS protocol

Destruction - Actively overwritten via Zeroize service

13. EAP-TLS Integrity Key (HMAC-SHA-1 160-bit):

Description - HMAC-SHA-1 key used for EAP-TLS integrity protection

Generation – N/A

Establishment – TLS v1.0 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 - EAP-TLS protocol

Destruction - Actively overwritten via Zeroize service

14. EAP-TLS Master Secret (384-bit):

Description - EAP-TLS shared secret (Master Secret)

Generation – N/A

Establishment – TLS v1.0 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 - EAP-TLS protocol
Destruction - Actively overwritten via Zeroize service

15. EAP-TLS Pre-Master Secret (2048-bit):
Description - EAP-TLS shared secret (Pre-Master Secret)
Generation – Approved SP800-90A DRBG; Approved as per FIPS 140-2 Annex C
Establishment – RSA key wrapped by the module during EAP-TLS session; allowed as per FIPS 140-2 IG D.9
Storage - Plaintext; stored in RAM
Entry – N/A
Output – Encrypted; RSA Key Wrapped with Server's Public Key
Key-To-Entity - EAP-TLS protocol
Destruction - Actively overwritten via Zeroize service

16. EAP-TLS Master Session Key (MSK) (256-bit):
Description – Used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0 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 - 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 client certificate 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 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
Output - N/A

Key-To-Entity - RADIUS server
Destruction - Actively overwritten via Zeroize service

19. EAP-TTLS Master Session Key (MSK) (256-bits):
Description - Used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0 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 - EAP-TTLS protocol
Destruction - Actively overwritten via Zeroize service

20. EAP-PEAP Master Session Key (MSK) (256-bits):
Description - Used as PMK for 802.11i KDF
Generation – N/A
Establishment – TLS v1.0 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 - EAP-PEAP protocol
Destruction - Actively overwritten via Zeroize service

14 Appendix B: Public Keys

The module supports the following public keys:

1. Client Diffie-Hellman Public Keys (2048-bit)
Description – Used in 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

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

3. Client EAP-TLS RSA Public Key used in the client certificate (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 TLS exchanges

4. Server EAP-TLS RSA Public Key used in the server certificate (RSA 2048-bit):
Description - Server certificate in EAP-TLS is used by the module to authenticate the server, used to RSA wrap the EAP-TLS Pre-Master Secret
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry - Plaintext; received from EAP-TLS server during TLS exchanges
Output - N/A

5. 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

6. Server EAP-TTLS RSA Public Key (RSA 2048-bit):
Description - Public key (RSA 2048-bit) for EAP-TTLS Server
Generation - N/A

Establishment – N/A
Storage - Plaintext; stored in RAM
Entry - Plaintext; received from EAP-TTLS server during TLS exchanges
Output - N/A

7. Server EAP-PEAP RSA Public Key (RSA 2048-bit):
Description - Public key (RSA 2048-bit) for EAP-PEAP Server
Generation - N/A
Establishment – N/A
Storage - Plaintext; stored in RAM
Entry - Plaintext; received from EAP-PEAP server during TLS exchanges
Output - N/A

8. 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

15 Revision History

Revision	Author	Description	Date: (DD.MM.YYYY)
04	Bill Dowd	Initial Release	25.11.2015