TI SimpleLink WiFi Networking Subsystem Crypto Module

Firmware Version 4.1.0.16

Hardware Chip ID: 0x311001

FIPS 140-2 Non-Proprietary Security Policy

Document Version 1.3

Last update: 2018-10-02

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1. Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for the TI SimpleLink WiFi Networking Subsystem Crypto Module version 4.1.0.16, Chip ID 0x311001. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 module.

The next sections describe the cryptographic module and how it conforms to the FIPS 140-2 specification in each of the required areas.

1.1. Description of Module

The SimpleLink CC3135 and CC3235 families are internet-on-a-chip Wi-Fi solutions that allow the connection of any low-cost, low power microcontroller unit (MCU) to the Internet of Things (IoT). It is a self-contained network processor with a dedicated ARM MCU (the ARM Cortex M4 for the CC3235, and a customer-choice MCU for the CC3135) and embedded TCP/IP stack that completely offloads Wi-Fi and internet protocols for the Host MCU. It consists of a Wi-Fi network processor subsystem, a Wi-Fi driver, multiple internet protocols in ROM, an ARM Cortex-M4 application microcontroller and peripherals.

Figure 1 demonstrates the physical look of the SimpleLink WiFi CC3135 and CC3235 family of chips.

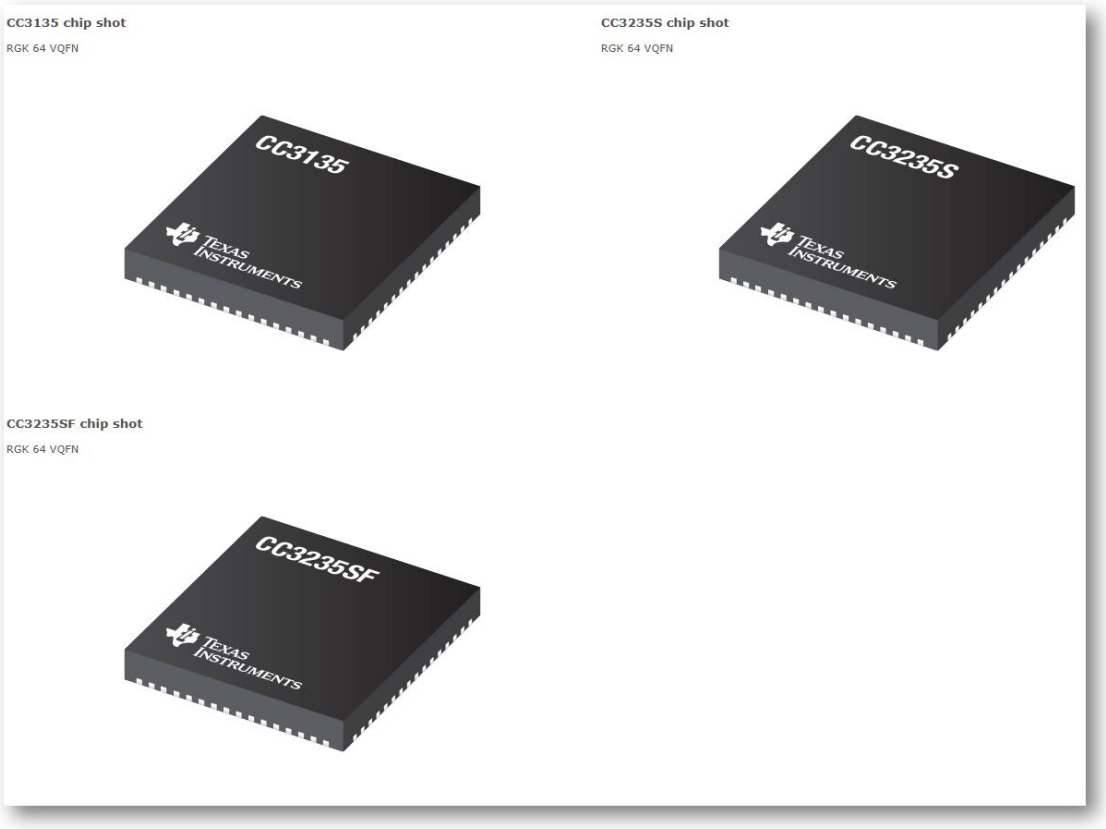

Figure 1: Physical representation of the CC3135 and CC3235 chips.

The TI SimpleLink WiFi Networking Subsystem Crypto Module (hereafter referred to as “the NWP module”, “the NWP” or “the module”) is a sub-chip cryptographic subsystem that resides within

SimpleLink CC3135 and CC3235 chips. The physical enclosure of these chips is the physical boundary of the NWP sub-chip module.

The Networking Subsystem Crypto Module is one of the two sub-chip modules on the same single chip subject to the FIPS 140-2 validation. It is bound to the other sub-chip module, TI SimpleLink WiFi MCU HW Crypto Engines Module, validated under FIPS 140-2 certificate #3272: this bound sub-chip module offers its HMAC and KDF algorithms for the integrity check procedure performed in the Service Pack of the Networking Subsystem Crypto Module during the power-on. The bound HW Crypto engine module also provides its AES and SHA/HMAC hardware engines for some services.

The module contains embedded hardware Triple-DES engine and firmware NWP code stored in ROM as well as the Service Pack (SP) NWP Record and FIPS configuration (FIPS Cfg) file stored in RAM after the extraction of the installation package. It provides the Wi-Fi connectivity coupled with TLS for communications security.

The logical boundary of the module consists of the orange blocks for the NWP (Figure 2 and Figure 3). The yellow blocks indicate the components belonging to the logical boundary of the bound sub-chip module, TI SimpleLink WiFi MCU HW Crypto Engines Module. Blocks of another color do not belong to any logical boundary.

SimpleLink CC31XX represents CC31 family chips including CC3135. Likewise, CC32XX includes CC3235 as a specific chip model within this family.

.

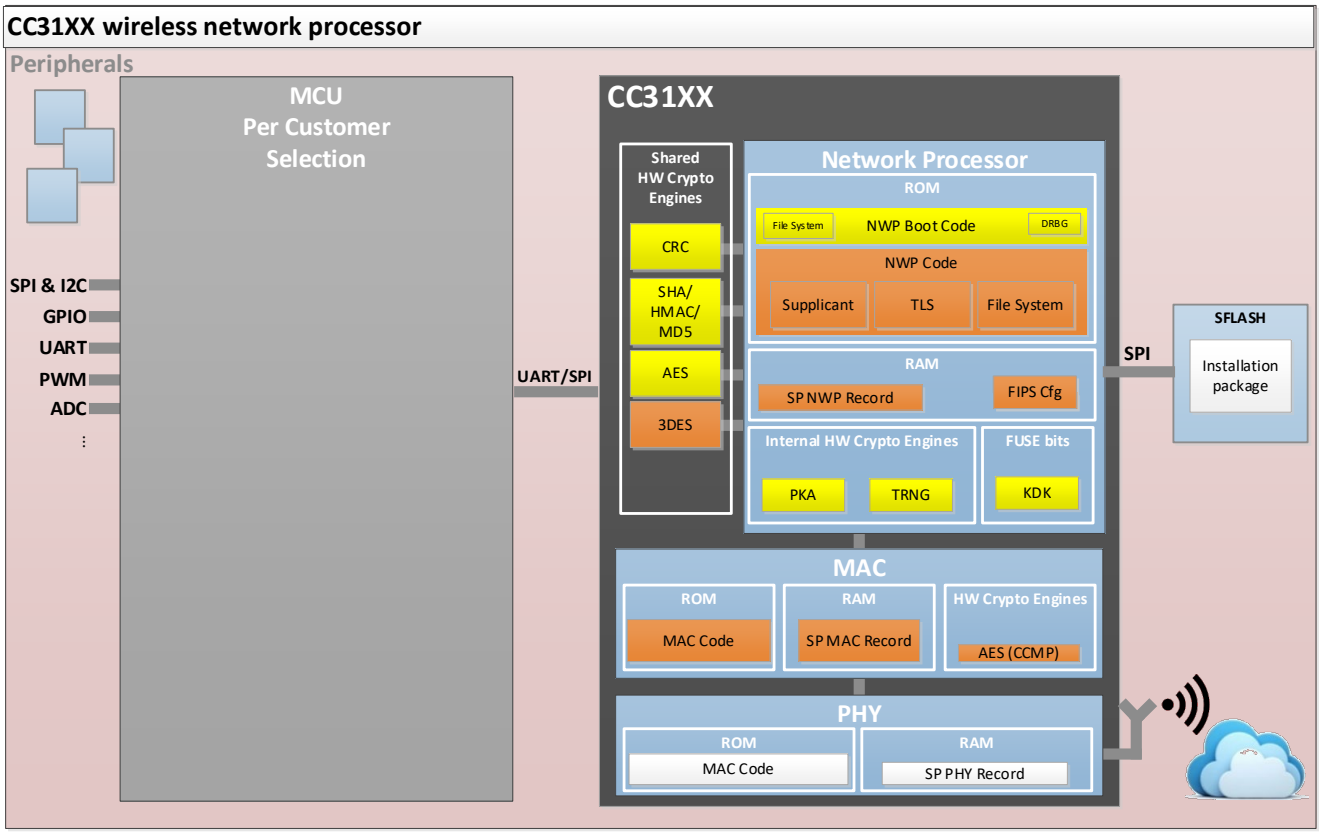

Figure 2: Logical boundary of the module on SimpleLink CC31XX chip.

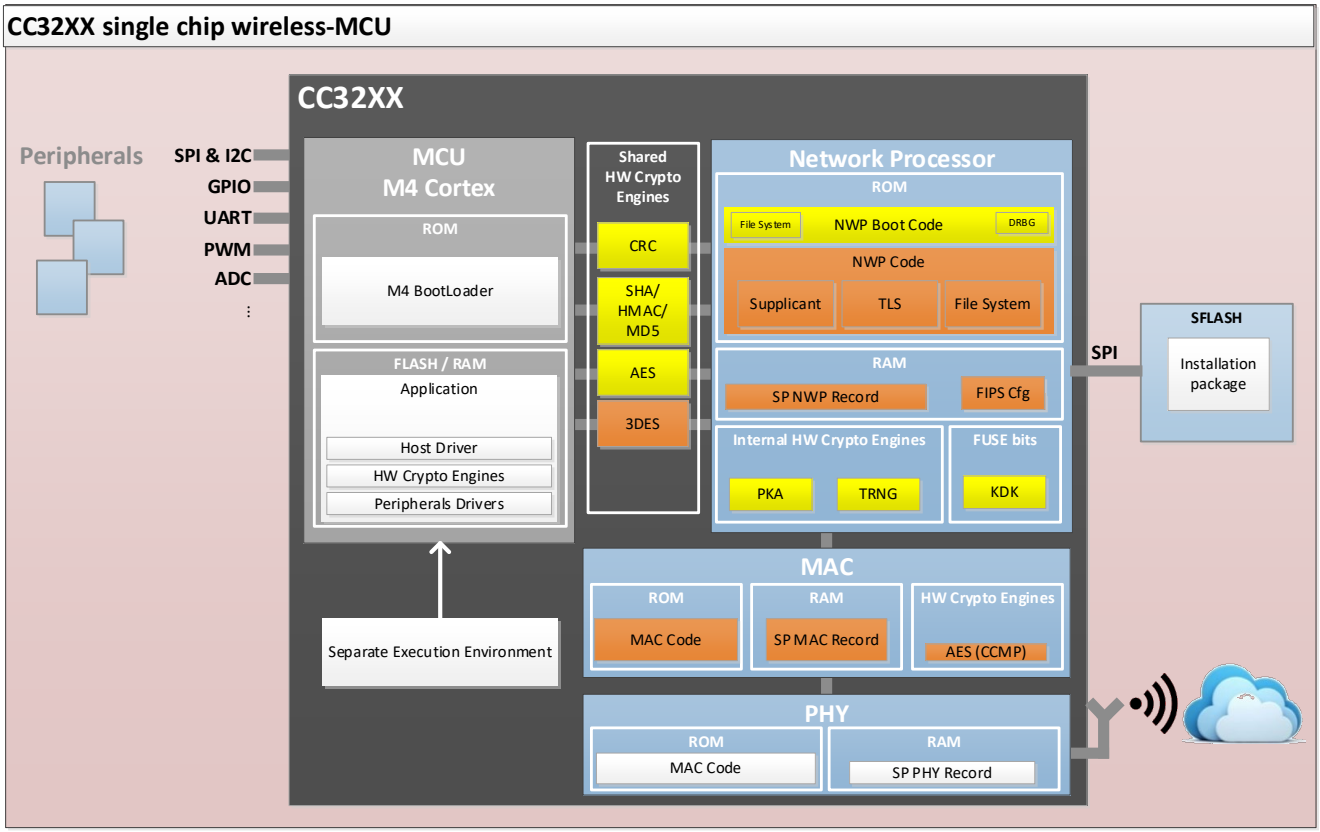

Figure 3: Logical boundary of the module on SimpleLink CC32XX chip.

The components within the logical boundary of the Networking Subsystem Crypto Module are listed in Table 1. The NWP Code implements wpa_supplicant, TLS and crypto libraries, as well as encryption to secure a file system. The actual files of the file system are stored on SFLASH memory, and this memory is outside of the module boundary. The wpa_supplicant uses AES-CCMP hardware crypto engine in the MAC hardware block to support WPA2 compliant to the IEEE 802.11 standard.

1.2. Version

The NWP module version is 4.1.0.16, Chip ID 0x311001. These numbers comprise all components of the module, including the Service Pack and FIPS Cfg file. Specifically, the second group of numbers (delimited by “.” from left to right) in the 4.1.0.16 version number uniquely identifies the Service Pack component. The Chip ID refers to the hardware chip component of the module and the FIPS Cfg file, wherein the 5 digits after the “0x” prefix identify the chip, and the last digit uniquely identifies the FIPS Cfg file.

Any extra digits after the 6 digits in the Chip ID are not relevant for the module version.

Table 1: Components of the NWP Cryptographic Module.

Component	Description
Triple-DES	Hardware component in Simple Link CC3135 and CC3235 chips
Network Processor	Hardware component in Simple Link CC3135 and CC3235 chips
NWP Code	Firmware in ROM of Network Processor on Simple Link CC3135 and CC3235 chips
SP NWP Record	Service Pack in RAM of Network Processor on Simple Link CC3135 and CC3235 chips that patches NWP code
FIPS Cfg	A configuration file in RAM of Network Processor on Simple Link CC3135 and CC3235 chips
MAC	Hardware component in Simple Link CC3135 and CC3235 chips
MAC Code	Firmware in ROM of MAC on Simple Link CC3135 and CC3235 chips
SP MAC Record	Service Pack in RAM of MAC on Simple Link CC3135 and CC3235 chips that patches MAC code
AES-CCMP	HW Crypto Engine in MAC on Simple Link CC3135 and CC3235 chips

1.3. FIPS 140-2 Validation

For the purpose of the FIPS 140-2 validation, the module is defined as a sub-chip hardware cryptographic module with a single chip embodiment validated at overall security level 1. Table 2 shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard.

Table 2: Security levels for each section of FIPS 140-2 standard.

FIPS 140-2 Section		Security Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services and Authentication	1
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
Overall Level		**1**

The module has been tested on the platforms specified in Table 3, which belong to the CC3135 and CC3235 family of chips.

Table 3: Tested platforms.

Test Platform (SoC Reference)	MCU
CC3135R	TI MSP430
CC3235S	ARM Cortex M4 (dedicated)
CC3235SF	ARM Cortex M4 (dedicated)

1.4. Modes of operation

The module supports two modes of operation:

- in "FIPS mode" (the FIPS Approved mode of operation), only approved or allowed security functions with sufficient security strength are offered by the module.
- in "non-FIPS mode" (the non-Approved mode of operation), non-approved security functions are offered by the module.

The module enters the operational mode after Power-On Self-Tests (POST) succeed. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength[1] of the cryptographic keys.

[1] See Section 5.6.1 in [SP 800-57] for a definition of “security strength”.

If the POST or the Conditional Tests fail (Section 7), the module goes into the error state. The status of the module can be determined by the availability of the module. If the module is available, then it had passed all self-tests. If the module is unavailable, it is because any self-test procedure failed and the module has transitioned to the error state.

Keys and Critical Security Parameters (CSPs) used or stored in FIPS mode shall not be used in non-FIPS mode, and vice versa.

2. Cryptographic Module Ports and Interfaces

The module provides cryptographic services and an application program interface (API). The physical ports are registers within the logical boundary of the sub-chip module. These registers hold the data for API parameters. The data flow in and out of registers via UART or SPI interfaces of the SimpleLink chip. Table 4 summarizes the four logical interfaces and their mappings to physical ports and interfaces:

Table 4: Ports and Interfaces.

Logical Interface	Physical Ports/Interfaces	Description
Data Input	Registers/UART/SPI	API input parameters for data
Data Output	Registers/UART/SPI	API output parameters for data
Control Input	Registers/Interrupts/UART/SPI	API function calls, API input parameters for control.
Status Output	Registers/Interrupts/UART/SPI	API return codes, API output parameters for status.
Power Input	Power Supply Port	Not applicable for the sub-chip module. The module receives power from the device in which the module is embedded.

The Data Input interface consists of the registers that hold the data for the input parameters of the API functions. The input data is received from the Serial Peripheral Interface (SPI) or Universal Asynchronous Receiver/Transmitter (UART) of the SimpleLink chip in which the sub-chip module resides.

The Data Output interface consists of registers that hold the data for the output parameters of the API functions. The output data leaves the physical boundary of the SimpleLink chip via its SPI or UART interfaces.

The Control Input interface consists of the API function calls and the input parameters used to control the behavior of the module. The API function calls are handled by the system scheduler as interrupts. The control input enters the registers of the sub-chip module via its SPI or UART interfaces.

The Status Output interface includes the return code of the API functions and the status sent through output parameters. The return code or status output may reside in the registers of the sub-chip module or be sent out of the chip physical boundary via its SPI or UART interfaces.

3. Roles, Services and Authentication

3.1. Roles

The module supports the following roles:

- **Crypto Officer role**: performs module installation and configuration.
- **User role**: performs all services (in both FIPS mode and non-FIPS mode of operation), except module installation and configuration.

The User and Crypto Officer roles are implicitly assumed by the entity accessing the module services. In other words, by invoking a specific service offered by the module, the role is implicitly assumed by the entity according to the service that was invoked by that entity.

3.2. Services in the FIPS Mode

The module provides services to users who assume one of the available roles. All services are described in detail in the user documentation.

Table 5 lists the Approved services and the non-Approved but allowed services in FIPS mode of operation. The table also lists the roles that can request the service, the algorithms involved with their corresponding CAVP certificate numbers (if applicable), and the Critical Security Parameters (CSPs) involved and how these CSPs are accessed.

Table 5: Cryptographic Services in FIPS mode of operation.

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
TLS and crypto libraries				
RSA digital signature verification	RSA #2916 (CC3135R) #2911 (CC3235S) #2913 (CC3235SF)	User	Read	RSA public key with 1024-bit (legacy, allowed by [SP800-131A]) and 2048-bit modulus sizes
	RSA Allowed by [SP800-131A]	User	Read	RSA public key with 4096-bit modulus size
ECDSA key generation	ECDSA #1445 (CC3135R) #1443 (CC3235S) #1444 (CC3235SF)	User	Write	ECDSA public and private key pair with curves P-256, P-384 and P-521
ECDSA signature generation	ECDSA #1445 (CC3135R) #1443 (CC3235S) #1444 (CC3235SF)	User	Write	ECDSA public and private key pair with curve P-256
ECDSA signature verification	ECDSA	User	Read	ECDSA public and private key

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
	#1445 (CC3135R) #1443 (CC3235S) #1444 (CC3235SF)			pair with curves P-256, P-384 and P-521
Message digest	SHA-1, SHA-256, SHA-384, SHA-512 #4360 (CC3135R) #4358 (CC3235S) #4359 (CC3235SF)	User	N/A	none
Message Authentication Code (MAC)	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 #3598 (CC3135R) #3596 (CC3235S) #3597 (CC3235SF)	User	Read	HMAC key of size at least 112 bits
Random Number Generation	Hash-DRBG #2124 (CC3135R) #2122 (CC3235S) #2123 (CC3235SF)	User	Read, Write	Seed (384-bit length) Internal state (V, C, key)
Key Wrapping/Encapsulation	RSA-based compliant to SP 800-56B (vendor affirmed per [FIPS140-2_IG] D.8)	User	Read, Write	Key encapsulation key (with modulus 2048 and 4096 bits) and encapsulated/ encrypted key
Key Agreement	KAS FFC CVL #1881 (CC3135R) CVL #1877 (CC3235S) CVL #1879 (CC3235SF) Prerequisite DSA #1398 (CC3135R) #1396 (CC3235S) #1397 (CC3235SF)	User	Read, Write	DH key pairs with $2048 \leq L \leq 15360$ and $224 \leq N \leq 512$ bits (security strength between 112 and 256 bits [SP800-57])
	KAS ECC CVL #1881 (CC3135R) CVL #1877 (CC3235S) CVL #1879 (CC3235SF) Prerequisite ECDSA #1445 (CC3135R)	User	Read, Write	ECDH key pairs with curves P-256, P-384 and P-521 (security strength between 128

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
	#1443 (CC3235S) #1444 (CC3235SF)			and 256 bits [SP800-57])
TLS Key Derivation	CVL #1882 (CC3135R) CVL #1878 (CC3235S) CVL #1880 (CC3235SF)	User	Read, Write	Key derivation key and derived keys
Key Derivation	SP 800-108 KDF in Counter Mode #212 (CC3135R) #207 (CC3235S) #209 (CC3235SF)	User	Read, Write	Key derivation key and derived keys, WPA2 pre-shared key (PSK), 802.11i KDF internal state, 802.11i Temporal Keys, 802.11i MIC keys (KCK), 802.11i Key Encryption Key (KEK), EAP-TLS MSK, EAP-TTLS MSK, EAP-PEAP MSK
NDRNG	N/A	User	Read, Write	Seed to DRBG (minimum 301 bits of entropy – Section 6.6)
TLS network protocol v1.0, v1.1 and v1.2 with the following cipher suites: TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_DH_anon_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	AES (ECB, CCM) #5435 (CC3135R) #5436 (CC3235S) #5438 (CC3235SF) AES (ECB, CBC, CTR, CFB, GCM, and CCM) from TI SimpleLink WiFi MCU HW Crypto Engines Module #5428 (CC3135R) #5429 (CC3235S) #5430 (CC3235SF) Triple-DES-CBC #2733 (CC3135R) #2731 (CC3235S) #2732 (CC3235SF)	User	Read, Write	AES or Triple-DES key, RSA public-private key, Shared Secret, Diffie-Hellman and EC Diffie-Hellman domain parameters and public-private keys, HMAC keys

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	SHA-1, SHA-256, SHA-384, SHA-512 #4360 (CC3135R) #4358 (CC3235S) #4359 (CC3235SF) RSA #2916 (CC3135R) #2911 (CC3235S) #2913 (CC3235SF) KAS FFC CVL #1881 (CC3135R) CVL #1877 (CC3235S) CVL #1879 (CC3235SF) Prerequisite DSA #1398 (CC3135R) #1396 (CC3235S) #1397 (CC3235SF) KAS ECC CVL #1881 (CC3135R) CVL #1877 (CC3235S) CVL #1879 (CC3235SF) Prerequisite ECDSA #1445 (CC3135R) #1443 (CC3235S) #1444 (CC3235SF)			
wpa_supplicant				
Key Wrapping	AES KW #5440 (CC3135R) #5437 (CC3235S) #5439 (CC3235SF)	User	Read, Write	Key wrapping key (128-bit) and wrapped key, 802.11i KEK, 802.11i GTK
Symmetric Encryption and Decryption	Triple-DES-CBC #2733 (CC3135R) #2731 (CC3235S) #2732 (CC3235SF)	User	Read	192-bit Three-key Triple-DES key
RSA digital signature verification	RSA #2915 (CC3135R)	User	Read	RSA public key with 1024-bit (legacy, allowed

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
	#2912 (CC3235S) #2914 (CC3235SF)			by [SP800-131A]) and 2048-bit modulus sizes
802.11i KDF	SP 800-108 KDF in Counter Mode #211 (CC3135R) #208 (CC3235S) #210 (CC3235SF)	User	Read, Write	Key derivation key and derived keys, WPA2 pre-shared key (PSK), 802.11i KDF internal state, 802.11i Temporal Keys, 802.11i MIC keys (KCK), 802.11i Key Encryption Key (KEK), EAP-TLS MSK, EAP-TTLS MSK, EAP-PEAP MSK
Key Pair generation (Finite Field)	DSA key pair #1398 (CC3135R) #1396 (CC3235S) #1397 (CC3235SF)	User	Read, Write	DH key pairs, L=2048, N=224, or L=2048, N=256
Provide wireless connection with Access Point with the establishment of a TLS tunnel: EAP EAP-TLS EAP-TTLS with MSCHAP EAP-TTLS with TLS EAP-TTLS with PSK EAP-PEAP0 with TLS EAP-PEAP0 with MSCHAP EAP-PEAP0 with PSK EAP-FAST AUTH PROVISIONING EAP-FAST UNAUTH PROVISIONING EAP-FAST NO PROVISIONING	In the context of a TLS connections, non-allowed algorithms (such as RC4, DES, HMAC-MD4, RSA-1024, Diffie-Hellman (512-bit, 1024-bit) used in wpa_supplicant for wireless connections (e.g., WEP and open connections) do not claim any security strength. Per IG 1.23, they are allowed to be used in the FIPS-mode. The security is provided by TLS tunnel over the wireless connection. The EAP methods provide the keys and CSPs that secure the TLS connection.	User	Read, Write	EAP-TLS keys/CSPs EAP-TTLS keys/CSPs EAP-PEAP keys/CSPs EAP-FAST keys/CSPs TLS KDF internal state (see individual EAP keys and CSPs descriptions in Table 7)
MAC hardware block				

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
AES CCMP Protocol	AES (ECB, CCM) #5435 (CC3135R) #5436 (CC3235S) #5438 (CC3235SF)	User	Read	128-bit AES keys (802.11i Temporal Keys, 802.11i GTK)
Triple-DES hardware (HW) block				
Symmetric Encryption and Decryption[2]	Triple-DES (ECB, CBC) #2725 (CC3135R) #2726 (CC3235S) #2727 (CC3235SF)	User	Read, Write	Three-key Triple-DES 192-bit key
From the Bound Module, TI SimpleLink WiFi MCU HW Crypto Engines Module				
Symmetric Encryption and Decryption	AES (ECB, CBC, CTR, CFB, GCM, and CCM) #5428 (CC3135R) #5429 (CC3235S) #5430 (CC3235SF)	User	Read, Write	AES key, used in TLS protocols
Message digest[3]	SHA-1, SHA-256 #4354 (CC3135R) #4355 (CC3235S) #4356 (CC3235SF)	User	n/a	none
Message Authentication Code[3] (MAC)	HMAC-SHA-1, HMAC-SHA-256 #3592 (CC3135R) #3593 (CC3235S) #3594 (CC3235SF)	User	Read	HMAC key of size at least 112 bits
Key Derivation[4]	Key Derivation Function in Counter Mode (KDF in CTR mode) [SP800-108]. #204 (CC3135R) #206 (CC3235S) #205 (CC3235SF)	Crypto Officer	Read, Write	Bound module uses KDK from its own boundary to derive HMAC keys to be used for self-test of the NWP module
Other FIPS 140-2 related services				
Show status	N/A	User	N/A	none

[2] This Triple-DES in the hardware (HW) block is not employed by the module for its upper level services. It is, however, available to the user of the module.
[3] Used for self-test of the NWP module only.
[4] Used when the Crypto Officer is installing the module, wherein the bound module derives HMAC keys from the KDK within its boundary. The process is described in Section 8.1.

Service	Algorithms, CAVP certificates	Role	Access to Keys/CSPs	Keys/CSPs
Self-Tests	N/A	User	Read	HMAC key (for module integrity test)
Zeroization	N/A	User	Write	All CSPs
Module Installation	N/A	Crypto Officer	Read, Write	none
Module Configuration	N/A	Crypto Officer	Read, Write	none

3.3. Services in the non-FIPS Mode

Table 6 lists the services only available in non-FIPS mode of operation. Using any of these services will implicitly turn the module into the non-FIPS mode of operation.

Table 6: Services in non-FIPS mode of operation.

Service	Role	Access	Keys and other Security Parameters
Symmetric encryption / decryption using Blowfish, Camellia, CAST, DES, IDEA, RC2, RC4, RC5, SEED.	User	Read	Symmetric keys
Asymmetric key generation using non-Approved key sizes.	User	Write	Public-Private key pairs (RSA, ECDSA)
Digital signature generation using non-Approved key sizes.	User	Read	Private keys (RSA, ECDSA)
Digital signature verification using non-Approved key sizes.	User	Read	Public keys (RSA, ECDSA)
Message digest using MD2, MD4, MD5, MDC-2, RIPEMD160.	User	N/A	none
MAC generation/verification using non-Approved keys.	User	Read	HMAC key less than 112-bit

Service	Role	Access	Keys and other Security Parameters
TLS network protocol v1.0, v1.1 and v1.2 with the following cipher suites: TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_MD5 TLS_DH_anon_WITH_RC4_128_MD5 TLS_DH_anon_WITH_DES_CBC_SHA SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_MD5 ECDHE_RSA_WITH_RC4_128_SHA TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	User	Read, Write	RSA key pairs, RC4 keys, DES keys

The following algorithms and key sizes are disallowed by NIST SP800-131A Rev1 and therefore can only be used in non-FIPS mode of operation. Thus, employing the following algorithms and key sizes will implicitly turn the module into the non-FIPS mode of operation.

- SHA-1 for digital signature generation.
- RSA keys with modulus size:
    - less than 2048 bits for digital signature generation.
    - less than 1024 bits for digital signature verification.
    - less than 2048 bits for key agreement and key transport schemes [SP800-56B].
- ECDSA keys with P-curve size:
    - less than 224 bits for digital signature generation.
    - less than 160 bits for digital signature verification.
- Key agreement using Diffie-Hellman and Menezes-Qu-Vanstone (MQV):
    - less than 2048 bits for length of *L* (public key) or less than 224 bits for length of *N* (private key).

3.4. Operator Authentication

There is no operator authentication; assumption of role is implicit by the used service(s).

4. Physical Security

The module is a sub-chip module implemented as part of the TI SimpleLink CC3135 and CC3235 family of chips, which include the tested platforms listed in Table 3. The TI SimpleLink family chip is a single chip with a production-grade enclosure and hence conforms to the Level 1 requirements for physical security.

5. Operational Environment

The module operates in a non-modifiable operational environment per FIPS 140-2 level 1 specifications. As such, the operational environment is considered as not applicable to the FIPS rules.

6. Cryptographic Key Management

Table 7 summarizes the keys and CSPs that are used by the cryptographic services implemented in the module.

Table 7: Life cycle of keys and Critical Security Parameters (CSPs).

Name	Generation/ Establishment	Entry/Exit	Storage	Usage	Zeroization
AES keys	Generated during the TLS handshake	Entered via API parameter. No exit.	RAM	File/data Encryption/ decryption	(Common to all keys and CSPs): Power off module RAM zeroization API (`sl_DeviceSet(SL_DEVICE_FIPS, SL_DEVICE_FIPS_ZEROIZATION, 0 , NULL);`)
Triple-DES keys	Generated during the TLS handshake	Entered via API parameter. No exit.	RAM	File/data Encryption/ decryption	
HMAC keys	Generated during the TLS handshake, or derived from the bound module's KDK	Entered via API parameter. No exit.	RAM	File/data integrity protection	
RSA private keys	From private key file in SFLASH	Entered via API parameter. No exit.	RAM and SFLASH, both outside the module boundary	Optional client-to-server authentication	
DSA private keys	From private key file in SFLASH	Entered via API parameter. No exit.	RAM and SFLASH, both outside the module boundary	Optional client-to-server authentication	
ECDSA private keys	From private key file in SFLASH	Entered via API parameter. No exit.	RAM and SFLASH, both outside the module boundary	Optional client-to-server authentication	
DH private keys	Output of DRBG as input to FIPS 186-4	N/A	RAM	TLS handshake	
ECDH private keys	Output of DRBG as input to FIPS 186-4	N/A	RAM	TLS handshake	
Seed	Generated by the NDRNG	N/A	RAM	Seed the SP 800-90A DRBG	
DRBG internal state (V, C, Key)	Generated by the DRBG	N/A	RAM	Generate random bit strings	

Name	Generation/ Establishment	Entry/Exit	Storage	Usage	Zeroization
DRBG output	Generated by the DRBG	No Entry. In the context of TLS, may be output in encapsulated form, e.g., encrypted by RSA key wrapping with TLS server's public key	RAM	Random bit strings. In the example, as an EAP-TLS Pre-Master Secret (PMS)	
WPA2 Pre-shared key (PSK)	N/A	Manually distributed, electronically entered in plaintext. No exit.	RAM and SFLASH, both outside the module boundary	Used for pre-shared key authentication and session key establishment, as well as for 802.11i KDF	
802.11i KDF Internal State	SP 800-108 KDF	N/A	RAM	Used for SP 800-108 KDF to calculate the WPA2 session keys	
802.11i Temporal Keys	SP 800-108 KDF	N/A	RAM	AES-CCM keys used for session encryption/ decryption	
802.11i MIC keys (KCK)	SP 800-108 KDF	N/A	RAM	Key confirmation keys (KCK) used for message authentication during session establishment	
802.11i Key Encryption Key (KEK)	SP 800-108 KDF	N/A	RAM	Used for AES Key Wrapping of the 802.11i Group Temporal Key (GTK)	
802.11i Group Temporal Key (GTK)	Established by key transport: wrapped with 802.11i KEK (AES 128-bit) ([FIPS140-2_IG] D.9)	Entered via key transport: wrapped with 802.11i KEK (AES 128-bit). No exit.	RAM	802.11i session key for broadcast communications	
TLS KDF Internal State	SP 800-135 KDF	N/A	RAM	Values of the TLS KDF internal state used in EAP-TLS, EAP-TTLS and EAP-PEAP	

Name	Generation/ Establishment	Entry/Exit	Storage	Usage	Zeroization
EAP-TLS Encryption Key	SP 800-135 KDF	N/A	RAM	AES-CBC key used to encrypt EAP-TLS session data	
EAP-TLS Integrity Key	SP 800-135 KDF	N/A	RAM	HMAC-SHA-1 key used for EAP-TLS integrity protection	
EAP-TLS Master Secret Key	SP 800-135 KDF	N/A	RAM	EAP-TLS shared secret (Master Secret)	
EAP-TLS Pre-Master Secret	Output of DRBG	Entry: N/A Exit: encrypted by RSA key wrapping with server's public key	RAM	EAP-TLS shared secret (Pre-Master Secret)	
EAP-TLS Master Session Key (MSK)	SP 800-135 KDF	N/A	RAM	Used as PMK for 802.11i KDF	
EAP-TTLS Master Session Key (MSK)	SP 800-135 KDF	N/A	RAM	Used as PMK for 802.11i KDF	
EAP-PEAP Master Session Key (MSK)	SP 800-135 KDF	N/A	RAM	Used as PMK for 802.11i KDF	

The following sections describe how cryptographic keys and other CSPs are managed during their life cycle.

6.1. Key Generation and Derivation

For generating ECDSA, DH and ECDH keys, the module implements asymmetric key generation services compliant with [FIPS186-4] and using a DRBG compliant to [SP800-90A]. A seed (i.e., the random value) used in asymmetric key generation is obtained from [SP800-90A] DRBG. In accordance with [FIPS140-2_IG] D.12, the cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per [SP800-133] (vendor affirmed).

Symmetric keys are derived from the shared secret established by Diffie-Hellman and EC Diffie-Hellman in a manner that is compliant to NIST SP 800-135 for TLS KDF.

6.2. Key Establishment

The module provides Diffie-Hellman and EC Diffie-Hellman key agreement schemes used as part of the TLS protocol key exchange. The module also provides Key Transport Methods, as approved by [FIPS140-2_IG] D.9. These methods include AES key wrapping per [SP800-38F] and RSA-based key encapsulation (using public key encryption and private key decryption primitives) as part of the TLS protocol and the 802.11i protocol as well.

Table 5 specifies the key sizes allowed in FIPS mode of operation. According to "Table 2: Comparable strengths" in [SP800-57], the key sizes of AES key wrapping, RSA, Diffie-Hellman and EC Diffie-Hellman provide the following security strength[5]:

- AES key wrapping provides 128 bits of security strength.
- RSA-based key encapsulation provides 112 or 150 bits of security strength.
- Diffie-Hellman key agreement provides between 112 and 256 bits of security strength.
- EC Diffie-Hellman key agreement provides between 128 and 256 bits of security strength.

The module supports key derivation for the TLS protocol. The module implements the pseudo-random functions (PRF) for TLSv1.0/1.1 and TLSv1.2.

6.3. Key Entry / Output

AES, Triple-DES keys, RSA private key file, DSA private key file, and ECDSA private key file may enter the module via API input parameters for encryption and decryption operations. The module does not support manual key entry or intermediate key output during the key generation process. In addition, the module does not produce key output in plaintext format outside its physical boundary.

6.4. Key / CSP Storage

Public and private keys are provided to the module by the calling process and are destroyed when released by the appropriate API function calls.

The module does not perform persistent storage of keys. The only exception is the HMAC key used for integrity test, which is stored in the module's file system. The HMAC key is used solely for the integrity check, and cannot be exported from the module or read by user APIs.

6.5. Key / CSP Zeroization

A general RAM zeroization API is provided: `sl_DeviceSet(SL_DEVICE_FIPS, SL_DEVICE_FIPS_ZEROIZATION, 0 , NULL)`. The API call zeroizes all the RAM, and thus zeroizes all the keys and CSPs.

Zeroization of all keys and CSPs in RAM can also be obtained by powering off the module, and then powering the module back on (power cycle).

The zeroization process results in a key or CSP being overwritten with zeroes.

6.6. Random Number Generation

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90A]. The output of DRBG is used as a context string for SP800-108 CTR KDF.

The DRBG implements a Hash_DRBG mechanism. The DRBG is initialized during module initialization and seeded by an on-chip Non-Deterministic Random Number Generator (NDRNG). The min-entropy estimate rate of this entropy source (per tested platform as indicated in Table 3) is demonstrated in Table 8. The length of the entropy_input string that forms the DRBG seed is 384 bits. Considering the lowest bit/bit entropy value in Table 8 (CC3135S), the NDRNG provides a 384-bit seed with at least 301 (truncated to an integer) bits of entropy to the DRBG during initialization (seed) and reseeding (reseed). The entropy of 301 bits is computed per the equation below.

$$MinEntropy = 0.785 \cdot 384 \cong 301$$

[5] See Section 5.6.1 in [SP 800-57] for a definition of "security strength".

The module performs continuous tests on the output of the NDRNG to ensure that consecutive random numbers do not repeat.

Table 8: Minimum entropy of the on-chip NDRNG per tested platform.

Test Platform (SoC Reference)	Min. Entropy (bits/byte)	Min. Entropy (bit/bit)
CC3135R	6.47	0.809
CC3235S	6.28	0.785
CC3235SF	6.46	0.808

7. Self Tests

7.1. Power-On Self-Tests (POSTs)

The module performs power-on tests automatically when the module is powered on; power-on tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the power-on tests, services are not available, and input and output are inhibited. The module's cryptographic services are not available until the power-o tests are completed and succeeded.

After the Power-On Self-Tests succeed, the module enters its Operational state. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength of the cryptographic keys.

If the POST fails, the module goes into the error state. The status of the module can be determined by the availability of the module. If the module is available, then it had passed all self-tests. If the module is unavailable, it is because the POST procedure failed and the module has transitioned to the error state.

7.1.1. Integrity Tests

The integrity test is performed in both the firmware/ROM contents of the module, and in the Service Pack/RAM portion.

The firmware/ROM code of the module is verified by comparing a CRC-16 value calculated at runtime with the checksum value stored in the module, and that was computed at build time.

The integrity of the RAM/Service Pack portion is verified by having the module compare separate HMAC-SHA-256 values (for sections of the Service Pack) calculated at runtime with the corresponding pre-calculated values stored in a file within the file system. The HMAC-SHA-256 algorithm utilized for this Service Pack integrity check is furnished by the bound module, TI SimpleLink WiFi MCU HW Crypto Engines Module. The HMAC key utilized in this integrity test is derived upon installation of the module (Section 8), and it is also done by the bound module and its [SP800-108] KDF service. Once derived, the HMAC key is stored in the module's file system.

7.1.2. Cryptographic algorithm tests

The module performs self-tests on all FIPS-Approved cryptographic algorithms supported in the approved mode of operation, using the known-answer tests (KAT) shown in Table 9.

Table 9: Self-Tests.

Algorithm	Test
AES	• KAT AES ECB, encrypt (wpa_supplicant implementation) • KAT AES ECB, decrypt (wpa_supplicant implementation) • KAT AES ECB, encrypt (MAC HW AES block implementation)
Triple-DES	• KAT Triple-DES CBC, encrypt (wpa_supplicant implementation) • KAT Triple-DES CBC, decrypt (wpa_supplicant implementation) • KAT Triple-DES ECB, encrypt (HW Triple-DES block implementation) • KAT Triple-DES ECB, decrypt (HW Triple-DES block implementation)
HMAC	• KAT HMAC-SHA-1

Algorithm	Test
	• KAT HMAC-SHA-256 • KAT HMAC-SHA-512
ECDSA	• KAT ECDSA (NIST P-256) signature generation • KAT ECDSA (NIST P-256) signature verification
RSA	• KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 signature verification (TLS and crypto libraries implementation) • KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 Encryption (TLS and crypto libraries implementation) • KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 Decryption (TLS and crypto libraries implementation) • KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 signature verification (wpa_supplicant implementation) • KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 Encryption (wpa_supplicant implementation) • KAT RSA 2048-bit key (PKCS#1 v1.5) with SHA-256 Decryption (wpa_supplicant implementation)
DRBG	• KAT Hash-DRBG
KAS ECC	• Primitive “Z” Computation KAT
KAS FFC	• Primitive “Z” Computation KAT

7.2. On-Demand self-tests

The on-demand Self-Test is achieved by power cycling. The self-tests initiated on demand perform the same cryptographic algorithm tests as those executed during power-on. While the on-demand self-tests are running, cryptographic services are not available and data output is inhibited.

7.3. Conditional Tests

The module performs conditional tests on the cryptographic algorithms per Table 10. If any of the conditional tests fail, the module goes into the error state and becomes unavailable as described in Section 7.1.

Table 10: Conditional Tests.

Algorithm	Test
ECDSA key generation	• Pair-wise consistency test
NDRNG	• Continuous test (previous and current random data are not equal)

Note: CRNGT on the SP800-90A DRBG is not required per IG 9.8 in [FIPS140-2_IG].

8. Guidance

8.1. Crypto Officer Guidance

In order to install the FIPS validated module, the subsequent steps must be followed:

- The chip and the serial flash must be physically assembled on the PCB.
- The chip must be programmed by installing an image with the Image Creator tool, both provided by the vendor. The image contains the FIPS140-2 installation package with the Service Pack, and programming must be done by checking the proper checkbox in the Image Creator tool to enable the FIPS Cfg file to be programmed. The image is signed by the vendor with an RSA-SHA-256 signature.
- The programming step is done automatically by the module, with the assistance of the bound module (TI SimpleLink WiFi MCU HW Crypto Engines Module). Upon the loading of the image and Service Pack (the signed installation package), the bound module verifies the package signature utilizing its own RSA signature verification services and stored public key. If the signature is successfully verified, the files are written to the SFLASH memory. The bound module then derives an HMAC key from its Key-Derivation Key (KDK – which is a key stored in that bound module's eFUSE storage during manufacture; see Figure 2 and Figure 3) and the output of its DRBG as context string and utilizes this key to compute HMAC-SHA-256 values for portions comprising the entirety of the Service Pack. The values are stored into the module's file system, and well as the HMAC key.
    - After the programming step is concluded, the module can be utilized normally. Subsequent power-ons of the module will always initiate the POST. The POST utilizes the bound module and its HMAC-SHA-256 service for integrity test on the Service Pack, using the stored HMAC values and key.

For more information on the programming tool, please refer to the Uniflash guide: http://www.ti.com/lit/pdf/swru469.

8.2. User Guidance

Upon the correct installation of the FIPS validated module, and success of the POST procedure, the module operates in either the FIPS mode, or the non-FIPS mode. The mode of operation is implicitly assumed depending on the security function invoked and the security strength of the cryptographic keys (Section 1.4).

8.2.1. AES-GCM IV

AES GCM encryption and decryption are used in the context of the TLS protocol version 1.2. The module is compliant with [SP 800-52] and the mechanism for IV generation is compliant with [RFC5288]. The operations of one of the two parties involved in the TLS key establishment scheme are performed entirely within the cryptographic boundary of the module, including the setting of the counter portion of the IV.

When the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module (acting as server or client) triggers a handshake to establish a new encryption key per Section 7.4.1.1 and Section 7.4.1.2 in [RFC5246] and compliant to [FIPS140-2_IG] A.5.

In case the module's power is lost and then restored, the key used for AES GCM encryption or decryption shall be re-distributed.

8.2.2. Triple-DES Keys

Data encryption using the same three-key Triple-DES key shall not exceed 2^{16} (64-bit) Triple-DES blocks, in accordance to [SP800-67] and IG A.13 in [FIPS140-2-IG].

8.2.3. Key Usage and Management

In general, a single key shall be used for only one purpose (e.g., encryption, integrity, authentication, key wrapping, random bit generation, or digital signatures) and be disjoint between the modes of operations of the module. Thus, if the module is switched between its FIPS mode and non-FIPS mode or vice versa (Section 1.4), the following procedures shall be observed:

- The DRBG engine shall be reseeded.
- CSPs and keys shall not be shared between security functions of the two different modes.

For more information please refer to:

- Product page: http://www.ti.com/product/CC3235

9. Mitigation of Other Attacks

The module does not implement security mechanisms to mitigate other attacks.

10. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The sub-chip module is not a standalone device. As a hardware component, it cannot be certified by the FCC. It is rather intended to be used within a larger device which would undergo standard FCC certification for EMI/EMC.

According to 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, the module is not subject to EMI/EMC regulations because it is a subassembly that is sold to an equipment manufacturer for further fabrication. That manufacturer is responsible for obtaining the necessary authorization for the equipment with the module embedded prior to further marketing to a vendor or to a user.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation Scheme
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
GTK	Group Temporal Key
HMAC	Hash Message Authentication Code
KAS	Key Agreement Schema
KAT	Known Answer Test
KCK	Key Confirmation Key
KDK	Key Derivation Key
KEK	Key Encryption Key
KW	AES Key Wrap
MAC	Message Authentication Code
MCU	Microcontroller Unit
NIST	National Institute of Science and Technology
NDRNG	Non-Deterministic Random Number Generator
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PMK	Pre-master Key
PR	Prediction Resistance

PSK	Pre-shared Key
RNG	Random Number Generator
RAM	Random Access Memory
ROM	Read Only Memory
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SPI	Serial Peripheral Interface
TDES	Triple-DES
TLS	Transport Layer Security
WPA	Wi-Fi Protected Access
UART	Universal Asynchronous Receiver/Transmitter

Appendix B. References

FIPS140-2 **FIPS PUB 140-2 - Security Requirements For Cryptographic Modules**
May 2001
http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

FIPS140-2_IG **Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program**
May 25, 2018
http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
March 2012
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**
Specifications Version 2.1
February 2003
http://www.ietf.org/rfc/rfc3447.txt

RFC3394 **Advanced Encryption Standard (AES) Key Wrap Algorithm**
September 2002
http://www.ietf.org/rfc/rfc3394.txt

RFC5246 **The Transport Layer Security (TLS) Protocol Version 1.2**
August 2008
https://tools.ietf.org/html/rfc5246

RFC5288 **AES Galois Counter Mode (GCM) Cipher Suites for TLS**
August 2008
https://tools.ietf.org/html/rfc5288

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38C **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf

SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-52	**Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations – Rev 1** April 2014 https://csrc.nist.gov/publications/detail/sp/800-52/rev-1/final
SP800-56Ar2	**NIST Special Publication 800-56A Revision 2 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** May 2013 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf
SP800-56B	**Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography** August 2009 http://csrc.nist.gov/publications/nistpubs/800-56B/sp800-56B.pdf
SP800-57	**NIST Special Publication 800-57 Part 1 Revision 4 - Recommendation for Key Management Part 1: General** January 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf
SP800-67	**NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** January 2012 http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf
SP800-90A	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-131A	**NIST Special Publication 800-131A Revision 1- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths** November 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf

SP800-133 **NIST Special Publication 800-133 - Recommendation for Cryptographic Key Generation**
December 2012
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133.pdf

SP800-135 **NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions**
December 2011
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf