Silvus Technologies, Inc.

SC4000 Series Mesh Radio

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

1.0 – General Information

1.1 Overview

This document defines the Security Policy for the Silvus Technologies SC4000 Series Mesh Radio, hereafter denoted the Module.

1.2 Security Levels

ISO/IEC 24759 Section 6.	FIPS 140-3 Section Title	Security Level
1	General	2
2	Cryptographic Module Specification	2
3	Cryptographic Module Interfaces	2
4	Roles, Services, and Authentication	2
5	Software/Firmware Security	2
6	Operational Environment	N/A
7	Physical Security	2
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	2
10	Self-tests	2
11	Life-cycle Assurance	2
12	Mitigation of other attacks	N/A
N/A	Overall	2

2.0 – Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Module is used in a family of MIMO radios used to provide a wireless mesh network. Each radio can operate in a multitude of configurations that are accessed via simple web pages within the radio. Settings such as transmit power, frequency, channel bandwidth, link adaptation and range control can be accessed by simply using a web browser to log into any radio within the network.

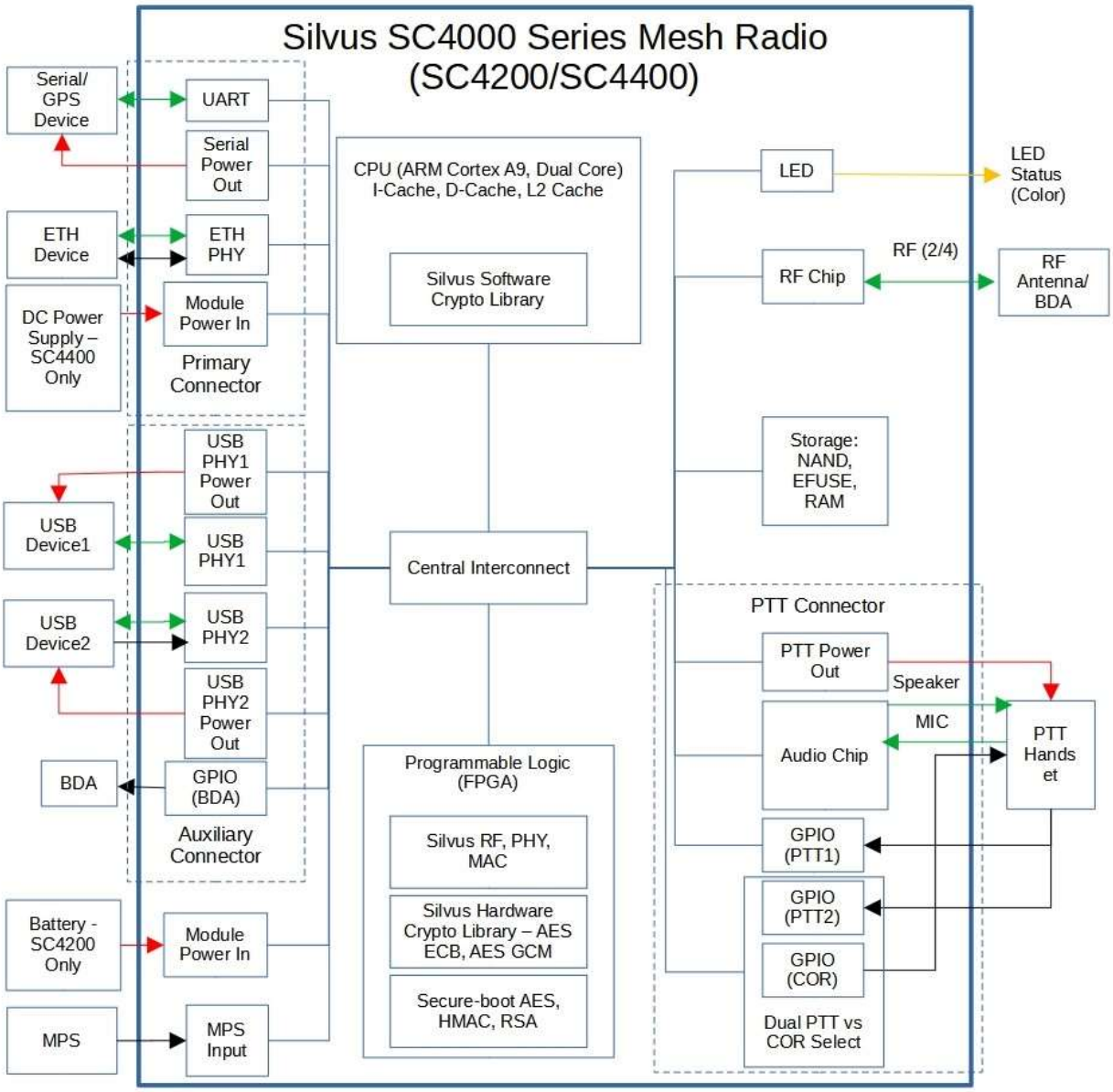
Module Type: Hardware

Module Embodiment: Multi-chip Standalone

Module Characteristics: None

Cryptographic Boundary:

The physical form of the Module is depicted in Figures 2 through 15. The module is composed of the baseboard circuit board surrounded by an opaque enclosure that serves as an EMI shield, heat sink, and FIPS enclosure. The Module is a multi-chip standalone embodiment with a limited operational environment. The cryptographic boundary is the surface of the entire module. The high-level block diagram showing the interconnections with external devices is shown below in Figure 1:


Silvus SC4000 Series Mesh Radio
(SC4200/SC4400)
Serial/
GPS
Device
UART
Serial
Power
Out
ETH
Device
ETH
PHY
DC Power
Supply –
SC4400
Only
Module
Power In
Primary
Connector
USB
PHY1
Power
Out
USB
Device1
USB
PHY1
USB
Device2
USB
PHY2
USB
PHY2
Power
Out
BDA
GPIO
(BDA)
Auxiliary
Connector
Battery -
SC4200
Only
Module
Power In
MPS
MPS
Input
CPU (ARM Cortex A9, Dual Core)
I-Cache, D-Cache, L2 Cache
Silvus Software
Crypto Library
Central Interconnect
Programmable Logic
(FPGA)
Silvus RF, PHY,
MAC
Silvus Hardware
Crypto Library – AES
ECB, AES GCM
Secure-boot AES,
HMAC, RSA
LED
LED
Status
(Color)
RF Chip
RF (2/4)
RF
Antenna/
BDA
Storage:
NAND,
EFUSE,
RAM
PTT Connector
PTT Power
Out
Speaker
MIC
Audio Chip
PTT
Hands
et
GPIO
(PTT1)
GPIO
(PTT2)
GPIO
(COR)
Dual PTT vs
COR Select

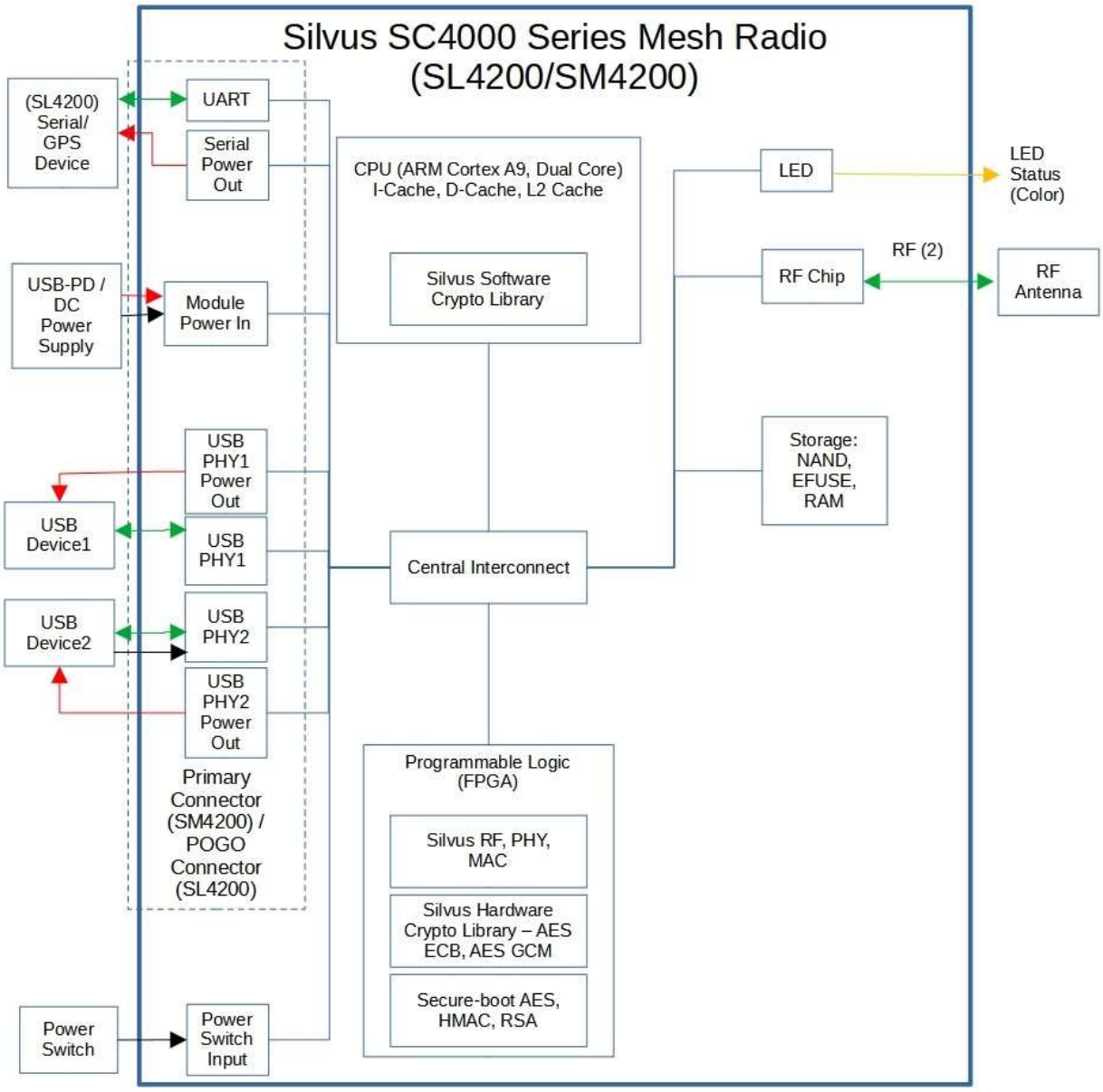


Figure 1 Silvus Crypto Module Block Diagram showing the crypto boundary and I/O with peripherals. The top figure is the SC4200/SC4400 and the bottom is the SL4200/SM4200.

Tested Operational Environment's Physical Perimeter (TOEPP):

SC4200

Figure 2 SC4200 showing two tamper evident seals.

Figure 3 SC4200 showing the serial number label.

Figure 4 SC4200 showing the battery input.

Figure 5 SC4200 showing the antenna, primary, auxiliary and PTT connectors.

SC4400

Figure 6 SC4400 showing one tamper evident seal.

Figure 7 SC4400 showing the second tamper evident seal.

Figure 8 SC4400 showing the serial number label.

Figure 9 SC4400 showing the antenna, primary, auxiliary and PTT connectors.

SL4200

Figure 10 SL4200 showing the serial number label.

Figure 11 SL4200 showing the two tamper evident seals.

Figure 12 SL4200 showing antenna and primary connector.

SM4200

Figure 13 SM4200 showing the serial number label.

Figure 14 SM4200 showing the two tamper evident seals.

Figure 15 SM4200 showing antenna and primary connector.

2.2 Version Information

The Module needs to be loaded with firmware version 4.1.0.0 to be considered FIPS 140-3 compliant. Any other firmware would render the Module non-compliant.

2.3 Operating Environments

Hardware Operating Environments:

Table 1. Operating Environments – Hardware

Model/Part Number(s)	Hardware Version(s)	Firmware Version(s)	Processor(s)	Non-Security Relevant Distinguishing Features (O)
SC4400/ SC4480E-235467-SBST	Rev B1	v4.1.0.0	Xilinx Zynq 7000 / ARM Cortex A9	Refer to Table 12
SC4200/ SC4240EP-235467-BB	Rev B7	v4.1.0.0	Xilinx Zynq 7000 / ARM Cortex A9	Refer to Table 12
SL4200/ SL4210-235-SB	Rev C3	v4.1.0.0	Xilinx Zynq 7000 / ARM Cortex A9	Refer to Table 12
SM4200/ SM4210-235-SB	Rev B2	v4.1.0.0	Xilinx Zynq 7000 / ARM Cortex A9	Refer to Table 12

The Module is designated as a limited operational environment under the FIPS 140-3 definitions. The Module includes a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-3 CMVP. Any other firmware loaded into this module is out of the scope of this validation and

requires a separate FIPS 140-3 validation. Table 1 shows the configurations that have been tested.

2.4 Excluded Components

N/A

2.5 Modes of Operation

Modes List and Description:

Table 5. Modes of Operation

Name	Description	FIPS	Status Indicator
Approved Mode	Single approved mode selected explicitly by setting the API “fips_mode” to [“1”] and going through the steps detailed below.	FIPS	“fips_mode” will return [“1”] “fips_ready” will return [“”] Alternatively, the web GUI shows a green-filled circle on the information side-panel.
Non Approved Mode	Single non-FIPS mode selected explicitly by setting the API “fips_mode” to [“”].	Non-FIPS	“fips_mode” will return [“0”] “fips_ready” will return [“”] Alternatively, the web GUI shows a red-filled circle on the information side-panel.

Mode change instructions and status indicators :

To operate in an Approved mode of operation the operator must perform the following steps. The module ships in a non-approved mode of operation.

1. Open the web interface and navigate to Security->Encryption tab.
2. Change the Approved mode to enable and click “Apply FIPS Mode”.
3. A popup window will appear, confirming this choice. Upon saying “OK”:
    a. All CSPs will be zeroized or reset to default values. All existing users will be deleted and three new users “basic”, “advanced” and “admin” with their respective roles will be auto-created with the default password - “HelloWorld”.
    b. The module will be subsequently rebooted into the Approved mode of operation; however, the module will not be fully in FIPS Approved mode unless the CO logs in and sets the CSPs to non-default values and configures some settings. The FIPS indicator on the informational side-panel will show an Amber sign. The Web GUI will now show all the CSPs and settings that need to be updated on a single page:
        i. Enable Password Complexity and set the minimum password length to be 8 characters.

ii. Set the login passwords for “basic”, “advanced” and “admin” to a new password that passes the following checks:
    1. Has a length greater than or equal to 8 characters.
    2. Has at least 1 lower case, 1 upper case, 1 special character and 1 number.
iii. Enable RF Link Encryption and set the API-Key, RF-Auth-Key, and RF-SK to non-default values and enable Encryption. The CO is required to input a previously unused RF-SK.
iv. Disable the SSH and SNMP service, and enable API Logs
v. Creates/Uploads a new TLS-Host-Priv, TLS-Host-Pub key pair.
vi. At this point, the module is fully operational in Approved mode. The user may change the Radio Mode to the Network Mode (0) setting to bring the module to operational state.

Once a module is in the Approved mode, the operator may put the module into the Non-Approved mode of operation by performing the following steps.

1. Open the web interface and navigate to Security->Encryption tab.
2. Change the Approved mode to disable and click “Apply FIPS Mode”.
3. A popup window will appear, confirming this choice. Upon saying “OK”:
    a. All CSPs will be zeroized or reset to default values. All existing users will be deleted and three new users “basic”, “advanced” and “admin” with their respective roles will be auto-created with the default password - “HelloWorld”.

The module will be subsequently rebooted into the Non-Approved mode of operation.

The user may check the status of the Approved mode and the firmware version in the following ways:
To confirm the FIPS firmware version and hardware model and part numbers, call the “build_information” API. The output should look like below:

SC4400:

{

"board_type":"SC4044",

"rfboard":"SN 2721 opt 21 range 2200~2500 4400~4940 ",

"rfboard1":"SN 2726 opt 21 range 2200~2500 4400~4940 ",

"digital_board":"SN 19112",

"model":"SC4400 Rev B1",

"build_date":"10-10-2019",

"build_tag":"streamscape_v4.1.0.0",

"phy_ts":"EX_1.12_0x6d8edf62d858be4a182dfac0bb54f497550af314_06-22-22, 11:02:10",

"part_no":"SC4480E-235467-SBST",

"entropy_version":"Silvus Clock Jitter Entropy Module v1.0"

}

SM4200:

{

"board_type":"SC4022",

"rfboard":"SN 55883 opt 5 range 2200~2500 ",

"rfboard1":"","digital_board":"SN 55883",

"model":"SM4200 Rev B2",

"build_date":"03-18-2022",

"build_tag":"streamscape_v4.1.0.0",

"phy_ts":"EX_1.12_0x33c956f734fe41896f03308bac2c79557d63809f_06-22-22, 09:53:18",

"part_no":"SM4210-235-SB",

"entropy_version":"Silvus Clock Jitter Entropy Module v1.0"},"id":2,"jsonrpc":"2.0"

}

SC4200:

{

"board_type":"SC4022",

"rfboard":"SN 2532 opt 21 range 2200~2500,4400~4940",

"rfboard1":"",

"digital_board":"SN 2916",

"model":"SC4200 Rev B7",

"build_date":"10-09-2019",

"build_tag":"streamscape_v4.1.0.0",

"phy_ts":"EX_1.12_0x33c956f734fe41896f03308bac2c79557d63809f_06-21-22, 19:18:38",

"part_no":"SC4240EP-235467-BB",

"entropy_version":"Silvus Clock Jitter Entropy Module v1.0"},"id":2,"jsonrpc":"2.0"

}

SL4200:

{

"board_type":"SC4022",

"rfboard":"SN 52918 opt 5 range 2200~2500 ",

"rfboard1":"",

"digital_board":"SN 52918",

"model":"SL4200 Rev C3",

"build_date":"02-10-2021",

"build_tag":"streamscape_v4.1.0.0",

"phy_ts":"EX_1.12_0x33c956f734fe41896f03308bac2c79557d63809f_06-22-22, 09:53:18",

"part_no":"SL4210-235-SB",

"entropy_version":"Silvus Clock Jitter Entropy Module v1.0"

}

The “model”, “part_no” and “build_tag” should match the corresponding fields in Table 1. This information is also present in the Build Info page on the Web GUI.

The Approved mode status can be checked using the API “fips_ready” and “fips_mode” and the GUI indicator. The “fips_mode” api returns [“1”] and the “fips_ready” returns [“”] if the module is fully in FIPS approved mode. The GUI indicator will show a green color.

If the Approved mode has been requested, but the FIPS initialization steps detailed above have not been completed, the “fips_mode” api returns [“1”] and the “fips_ready” returns a JSON array of the API that need changes, e.g. [“enc_rf_auth_key”] if the RF-Auth-Key is not set to a non-Factory value. The GUI indicator on the side panel will show an amber color.

If the Approved mode is disabled, the “fips_mode” api returns [“0”]. The GUI indicator will show a red color.

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.6 Approved Algorithms

Table 6. Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/Method	Description/Key Size(s)/Key Strength(s)	Use/Function
SL4200 & SM4200 Only				
A3420	AES SP 800-38A	ECB Encrypt	256 bits	Encrypt
A3420	AES SP 800-38D	GCM Encrypt, Decrypt	256 bits	Encrypt/Decrypt
SC4200 & SC4400 Only				
A3421	AES SP 800-38A	ECB Encrypt	256 bits	Encrypt
A3421	AES SP 800-38D	GCM Encrypt, Decrypt	256 bits	Encrypt/Decrypt
All Models				

A3422	AES SP 800-38A	CTR Encrypt, Decrypt	256 bits	Encrypt/Decrypt
A3422	AES SP 800-38D	GCM Encrypt, Decrypt	256 bits	Encrypt/Decrypt
A3422	DRBG SP 800-90Ar1	CTR with DF	AES-256	Deterministic Random Number Generation
A3422	ECDSA FIPS 186-4	KeyGen	P-256, P-384, P-521	KeyGen for KAS-ECC-SSC
A3422	ECDSA FIPS 186-4	KeyVer	P-256, P-384, P-521	KeyVer for KAS-ECC-SSC for RF link only.
A3422	ECDSA FIPS 186-4	SigGen	P-256, P-384, P-521 with SHA-384	SigGen for TLS v1.3
A3422	ECDSA FIPS 186-4	SigVer	P-256, P-384, P-521 with SHA-384. P-521 with SHA-256.	SigVer for TLS v1.3 and Firmware Load Test.
E25	ENT(NP) SP 800-90B	N/A	512 bits from entropy source. Full entropy.	Used to seed SP 800-90A DRBG. Entropy source provides 1 bit of entropy per bit output. Entropy source provides at least 256 bits of security.
A3422	HMAC FIPS 198-1	HMAC-SHA2-256, HMAC-SHA2-384	Key size: 256-bit	Message Authentication, KDF Primitive.
A3422	KAS-ECC-SSC 56Ar3	Ephemeral Unified Responder/Initiator	P-256, P-384, P-521	Key agreement for TLS and RF Link. Key establishment methodology provides between 128 and 256 bits of encryption strength.
A3422	KDA One Step SP-800 56Cr1	Fixed Info: uPartyInfo || vPartyInfo	SHA2-256	Key derivation after KAS-ECC-SSC for RF link.
A3422	KDA HKDF 56Cr1	Fixed Info: uPartyInfo || vPartyInfo,	HMAC SHA2-256	TLS v1.3
A3422	KTS SP 800-38F	AES-GCM	256-bit	Key Wrapping / Unwrapping
A3422	SHS FIPS 180-4	SHA2-256, SHA2-384	N/A	Message Digest Generation, Password Obfuscation.
A3225	SHS FIPS 202	SHA3-256	N/A	Vetted conditioner

Table 7. Vendor Affirmed Algorithms

Algorithm	Algorithm Properties	OE	Reference
CKG IG D.H	Section 4 Section 5.2 Section 6.1 Section 6.2.1	Xilinx Zynq 7000 / ARM Cortex A9	SP 800-133 Rev 2 IG D.H

NOTE: Module does not support Non-Approved Algorithms Allowed in the Approved Mode of Operation; see Table 8 for “No security claimed” algorithms.

Table 8. Non-Approved, Allowed Algorithms with No Security Claimed

Algorithm	Caveat	Use/Function
AES	No security claimed	For WPA2 controlled 802.11. This algorithm is not used whatsoever to meet any FIPS 140-3 requirements. This algorithm does not access or share CSPs used during an approved service. The algorithm is redundant to an approved algorithm. The module uses TLS v1.3 to encrypt all configuration traffic, hence the use of AES is redundant. The use of this algorithm is unambiguous and cannot be easily confused for a security function. As per FIPS 140-3 IG 2.4.A, TLS V1.3 is a secure channel operated over an insecure communications channel (WPA2).
KDF	No security claimed	PBKDF2 for WPA2. This algorithm is not used whatsoever to meet any FIPS 140-3 requirements. This algorithm does not access or share CSPs used during an approved service. The algorithm is redundant to an approved algorithm. The module uses TLS v1.3 to encrypt all configuration traffic, hence the use of PBKDF2 is redundant. The use of this algorithm is unambiguous and cannot be easily confused for a security function. As per FIPS 140-3 IG 2.4.A, TLS V1.3 is a secure channel operated over an insecure communications channel (WPA2).
RSA [1224]	No security claimed	RSA SigVer used for Secure boot. Validated by chip manufacturer but not on this OE. The RSA algorithm is not used whatsoever to meet FIPS 140-3 requirements. The algorithm does not access or share CSPs used during an approved service. The RSA algorithm is not intended to be used as a security function as the module already satisfies the firmware integrity test requirements using 32-bit CRC. The use of this algorithm is unambiguous and cannot be easily confused for a security function.
SHS [2034]	No security claimed	For SHS used for RSA SigVer used for Secure boot. Validated by chip manufacturer but not on this OE. The SHS algorithm is not used whatsoever to meet

		FIPS 140-3 requirements. The algorithm does not access or share CSPs used during an approved service. The SHS algorithm is not intended to be used as a security function as the module already satisfies the firmware integrity test requirements using 32-bit CRC. The use of this algorithm is unambiguous and cannot be easily confused for a security function.
AES [2363]	No security claimed	For AES-CBC 256-bit firmware decryption by Secure boot. Validated by chip manufacturer but not on this OE. The AES algorithm is not used whatsoever to meet FIPS 140-3 requirements. The algorithm does not access or share CSPs used during an approved service. The AES algorithm is not intended to be used as a security function, FIPS 140-3 does not require encrypted firmware at rest. As such, the firmware is being treated as plaintext and the module already satisfies the firmware integrity test requirements using 32-bit CRC. The use of this algorithm is unambiguous and cannot be easily confused for a security function.
HMAC [1465]	No security claimed	For HMAC-SHA-256 firmware authentication by Secure boot. Validated by chip manufacturer but not on this OE. The HMAC algorithm is not used whatsoever to meet FIPS 140-3 requirements. The algorithm does not access or share CSPs used during an approved service. The HMAC algorithm is not intended to be used as a security function as the module already satisfies the firmware integrity test requirements using 32-bit CRC. The use of this algorithm is unambiguous and cannot be easily confused for a security function.

Table 9. Non-Approved, Not Allowed Algorithms

Algorithm	Use/Function
DES-ECB	DES block encryption of RF data packets in the non-Approved mode. Also used for SNMP in non-Approved mode.
AES-ECB	AES block encryption of RF data packets in the non-Approved mode.
MD5	Used for SNMP in non-Approved mode.
SNMP KDF	Used for SNMP in non-Approved mode.
SSH KAS - ECDHE P-521	SSH key agreement.
SSH KDF – SHA 512	SSH KDF.
SSH Cipher – AES CTR 256-bit	SSH Data Encryption.
SSH Integrity – HMAC-SHA-256	SSH Data Integrity.

2.7 Security Function Implementations

Table 10. Security Function Implementations

Name	Type	Description	SF Properties	Algorithms	Algorithm Properties
KAS SSC (TLS)	KAS	SSP Establishment for TLS	Provides between 128 and 256 bits of encryption strength.	KAS-SSC [A3422]	P-256 P-384 P-521
KAS SSC (RF)	KAS	SSP Establishment for RF Link	Provides 256 bits of encryption strength.	KAS-SSC [A3422]	P-521
KTS	KTS	TLS Key Transport	Provides 256 bits of encryption strength.	AES-GCM [A3422]	256-bit

2.8 Algorithm Specific Information

Note: no parts of TLS v1.3 other than the approved cryptographic algorithms and the KDFs have been tested by CAVP and CMVP.

AES-GCM IV Generation

The module offers two AES GCM implementations: a) AES-GCM 256-bit used by TLS v1.3. b) AES-GCM 256-bit used by the RF Wireless Link service.

TLS v1.3:

For TLS 1.3, the module implements the TLS_AES_256_GCM_SHA384 cipher as defined in RFC 8446 Appendix B.4. It uses the context of Scenario 5 of IG C.H.

The module implements, within its boundary, a 64-bit counter for IV generation. If the IV exhaustion condition is observed, which will trigger a session termination or a re-key due to session re-establishment.

In the event the module's power is lost and restored, all previous TLS 1.3 session state is lost, hence a re-key due to session re-establishment will automatically enforce the IV uniqueness requirements.

RF Wireless Link:

There are two configuration options: a) AES-GCM 256-bit using keys generated by KAS-ECC-SSC b) AES-GCM 256-bit using a static key (RF-SK). For both cases, the IV is 96-bit, with 32 bits fixed to the unique 32-bit serial number of the module, and 48-bits dedicated to a 48-bit counter starting from 0. This follows IG C.H Scenario 3. The remaining 16 bits are fixed. If IV exhaustion is detected, the module will shut down the wireless interface and require zeroization

and subsequent FIPS initialization as described in Section 2.5. During FIPS initialization, the CO is required by policy to input a previously unused static RF-SK.

AES-GCM 256-bit using KAS-ECC-SSC:

If this option is chosen, the module will implement 56Ar3 compliant KAS-ECC-SSC for key agreement. In the event the module power is lost and restored, the key agreement is restarted, resulting in a new key.

AES-GCM 256-bit using Static Key:

If this option is chosen, the CO will input the 256-bit key to be used. Every 5 minutes, the module will record the last 48-bit counter used over the previous 5 minutes into non-volatile storage. In the event the module power is lost and restored, the module will resume the counter from the last stored value, but it will add an increment to ensure the next IV will be unique as required by 38D Section 9.1.

2.9 RNG and Entropy

Table 11. Entropy Sources

Name	Type	Operating Environment	Sample Size	Entropy Per Sample	Conditioning Components (CAVP number if vetted)
Silvus Clock Jitter Entropy Module #E25	Non-Physical	Xilinx Zynq Dual Core A9 CPU	8 bits	Full entropy, 256 bits of min entropy per 256-bit output block.	SHA-3 (#A3225)

Entropy Information: The Silvus Clock Jitter Entropy Module relies on small differences in CPU execution times (clock jitter) as an entropy source. It resides within the overall module and supplies entropy to seed the AES CTR DRBG described below. The module complies with Section 4 of the Public Use Document (https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E25_PublicUse.pdf).

RNG Information: The module uses the AES CTR DRBG seeded by the entropy source above as its RNG source. The RNG source is used for the following:

1. Generating the SSP Cookie-Key
2. ECDSA/ECDHE KeyGen

2.10 Key Generation

The module implements the below sections from SP800-133r2 compliant to CKG IG D.H for cryptographic key generation:

Section 4: Using the Output of a Random Bit Generator.

Section 5.2: Key Pairs for Key Establishment
For ECDSA and ECDHE keys.
Section 6.1: Direct Generation of Symmetric Keys
For SSP Cookie-Key.
Section 6.2.1: Symmetric Keys Generated using a Key Agreement Scheme
For KDA for TLS v1.3 and RF Wireless Link.

2.11 Key Establishment

The module implements SP-800 56Ar3 compliant KAS-ECC-SSC for key agreement for TLS v1.3 and RF Wireless Link. It complies with FIPS 140-3 IG D.F, Scenario 2, Path (2), whereby KAS-ECC-SCC and KDA were CAVP tested. See Section 2.6 above.

The module implements the TLS v1.3 cipher suite TLS_AES_256_GCM_SHA384, utilizing the AES-GCM portion for KTS.

2.12 Industry Protocols

The module uses the following industry standard protocols which use cryptography:

- TLS v1.3
- SNMP v1, v2, v3 (Only in the Non-Approved mode).
- WPA2 (No security claimed).
- SSHv2 (Only in the Non-Approved mode)

2.13 Design and Rules

Upon power-on, and after the pre-operational self-tests are successfully concluded, the module automatically transitions to the operational state.

2.14 Initialization

Please see Section 2.5.

2.15 Additional Information

N/A

3.0 Cryptographic module interfaces

3.1 Ports and Interfaces

The module’s ports and associated logical interface categories are listed in Table 12.

Table 12. Ports and Interfaces

Physical Port	Logical Interface	Data that passes over the port/interface
SC4200		
Battery (Qty 1)	Power Input	N/A
Primary Port (Qty 1)	Ethernet (Qty 1): Data I/O, Control I/O.	User data and module configuration encapsulated as IEEE 802.3 Ethernet frames.
	Serial (Qty 1): Data I/O, Power Output.	User data and module diagnostics encapsulated as RS232 serial data.
Auxiliary Port (Qty 1)	USB1 (Qty 1): Data I/O, Power Output (VBUS).	With USB-Ethernet adapter, OR USB-WiFi adapter: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. With USB-Serial adapter or GPS: User data encapsulated as RS232 serial data. With USB-Storage: Silvus signed license files to perform unauthenticated zeroize.
	USB 2 - OTG (Qty 1): Data I/O, Power Output (VBUS-Host Mode Only), Control Input.	Operates like USB1 in Host Mode. With USB-RNDIS Host and USB 2 in Client Mode: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. In this mode, the module will not do power output (VBUS).
	BDA GPIO (Qty 1): Control Output.	Controls an external Bi-Directional Amplifier (BDA).
PTT Port (Qty 1)	Power Output (Qty 1).	N/A
	Speaker (Qty 1): Data Output.	Analog audio.
	MIC (Qty 1): Data Input.	Analog audio.
	PTT GPIO1 (Qty 1): Control Input	N/A
	PTT GPIO2 (Qty 1 – Control Input) OR PTT COR GPIO (Qty 1 – Control Output).	N/A
LED (Qty. 1)	Status Output	N/A
Multi-Position Switch (Qty. 1)	Control Input	N/A

RF (Qty. 2)	Data Input/Output	User data and module configuration encapsulated as Silvus MIMO Waveform frames.
SC4400		
Primary Port (Qty 1)	Ethernet (Qty 1): Data I/O, Control I/O.	User data and module configuration encapsulated as IEEE 802.3 Ethernet frames.
	Serial (Qty 1): Data I/O, Power Output.	User data and module diagnostics encapsulated as RS232 serial data.
	DC Power Input (Qty 1).	N/A.
Auxiliary Port (Qty 1)	USB1 (Qty 1): Data I/O, Power Output (VBUS).	With USB-Ethernet adapter, OR USB-WiFi adapter: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. With USB-Serial adapter or GPS: User data encapsulated as RS232 serial data. With USB-Storage: Silvus signed license files to perform unauthenticated zeroize.
	USB 2 - OTG (Qty 1): Data I/O, Power Output (VBUS-Host Mode Only), Control Input.	Operates like USB1 in Host Mode. With USB-RNDIS Host and USB 2 in Client Mode: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. In this mode, the module will not do power output (VBUS).
	BDA GPIO (Qty 1): Control Output.	Controls an external Bi-Directional Amplifier (BDA).
PTT Port (Qty 1)	Power Output (Qty 1).	N/A
	Speaker (Qty 1): Data Output.	Analog audio.
	MIC (Qty 1): Data Input.	Analog audio.
	PTT GPIO1 (Qty 1): Control Input	N/A
	PTT GPIO2 (Qty 1 – Control Input) OR PTT COR GPIO (Qty 1 – Control Output).	N/A

LED (Qty. 1)	Status Output	N/A
Multi-Position Switch (Qty. 1)	Control Input	N/A
RF (Qty. 4)	Data Input/Output	User data and module configuration encapsulated as Silvus MIMO Waveform frames.
SL4200		
POGO Port (Qty 1)	USB1 (Qty 1): Data I/O, Power Output (VBUS).	With USB-Ethernet adapter, OR USB-WiFi adapter: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. With USB-Serial adapter or GPS: User data encapsulated as RS232 serial data. With USB-Storage: Silvus signed license files to perform unauthenticated zeroize.
	USB 2 - OTG (Qty 1): Data I/O, Power Output (VBUS-Host Mode Only), Control Input.	Operates like USB1 in Host Mode. With USB-RNDIS Host and USB 2 in Client Mode: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. In this mode, the module will not do power output (VBUS).
	Serial (Qty 1): Data I/O, Power Output.	User data and module diagnostics encapsulated as RS232 serial data.
	USB-PD (Qty 1): Power Input, Control Input.	N/A
	DC Power Input (Qty 1).	N/A
LED (Qty. 1)	Status Output	N/A
Power Switch (Qty. 1)	Control Input	N/A
RF (Qty. 2)	Data Input/Output	User data and module configuration encapsulated as Silvus MIMO Waveform frames.
SM4200		
Primary Port (Qty 1)	USB1 (Qty 1): Data I/O, Power Output (VBUS).	With USB-Ethernet adapter, OR USB-WiFi adapter: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames.

		With USB-Serial adapter or GPS: User data encapsulated as RS232 serial data. With USB-Storage: Silvus signed license files to perform unauthenticated zeroize.
	USB 2 - OTG (Qty 1): Data I/O, Control Input.	Operates like USB1 in Host Mode. With USB-RNDIS Host and USB 2 in Client Mode: User data and module configuration encapsulated as IEEE 802.3 Ethernet frames. In this mode, the module will not do power output (VBUS).
	USB-PD (Qty 1): Power Input, Control Input.	N/A
	DC Power Input (Qty 1).	N/A
LED (Qty. 1)	Status Output	N/A
Power Switch (Qty. 1)	Control Input	N/A
RF (Qty. 2)	Data Input/Output	User data and module configuration encapsulated as Silvus MIMO Waveform frames.

The module LED has the following color/pattern mapping:

- Solid Red: Module powering up. If the module spends more than 2 minutes in this state, it is bricked and needs factory repair.
- Blinking Red: This may be caused by the following:
    - When configured for the approved mode, this may indicate that the module needs further configuration to be fully in the approved mode (e.g. setting CSPs to non-default values). The user needs to check the FIPS indicator to verify this. If the module is already fully in the approved mode, this could indicate that the module is in FIPS error state.
    - If the user has run the “Spectrum Scan” diagnostic, the LED will blink red until the spectrum scan has completed.
    - On the SC4200, a rapidly blinking red for 1s indicates that the battery charge has fallen below 20% capacity.
    - If the user has set the MPS to position “Z” and power cycled the module, the module will power up with LED solid red, and then blinking green as usual. Then the zeroization process begins. Once the zeroization is completed, the LED will blink red.
- Blinking Green: Module fully powered up, but not connected to any peer module over the RF link. When configured for FIPS approved mode, this indicates that the module is ready, and all services are up.

- Solid Green: Module fully powered up and connected to at least one peer module over the RF link. When configured for FIPS approved mode, this indicates that the module is ready, and all services are up.
- Rapidly Flashing Green: On SC4200 and SC4400, when the MPS position is changed, the LED will rapidly flash green until the action mapped to the new MPS position has been completed.

3.3 Control Interface Not Inhibited

The control output for BDA and PTT will be turned off when in FIPS error state. Control output for USB and Ethernet will be left on to allow the user to check the web GUI and confirm that the module is in FIPS error state. This is required since the module only has an LED interface for status output.

4.0 Roles, services, and authentication

4.1 Authentication Methods

Table 13. Authentication Methods

Name	Description	Mechanism	Strength Each Attempt	Strength Per Minute
Password	Passwords are minimum 8 alphanumeric characters (a-z, A-Z, 0-9).	SHA-384 hash verification.	62 different values per character. Probability of guessing the password in a single attempt is 1/(62^8).	Three login attempts in a minute are allowed before the module locks out the user for 1 minute, so the strength per minute is 3/(62^8).
Cookie	Cookies are authenticated using HMAC-SHA-384 with the 256-bit Cookie-Key.	HMAC-SHA-384 hash verification.	1/(2^256)	Due to CPU limitations, the module can perform 20 authentications per second, or 1200 per minute. So the strength per minute is 1200/(2^256).

API Key	The API is authenticated using HMAC-SHA-256 with the 256-bit API-Key.	HMAC-SHA-256 hash verification.	1/(2^256)	Due to CPU limitations, the module can perform 20 authentications per second, or 1200 per minute. So the strength per minute is 1200/(2^256).
RF Link	The KAS messages are authenticated using HMAC-SHA-256 with the 256-bit RF-Auth-Key.	HMAC-SHA-256 hash verification.	1/(2^256)	Due to CPU limitations, the module can perform 20 authentications per second, or 1200 per minute. So, the strength per minute is 1200/(2^256).

4.2 Roles

The module supports role-based authentication with four distinct operator roles, Cryptographic Officer (Admin), Advanced, Basic and Mesh Node. The cryptographic module enforces the separation of roles using a standard session-based architecture. Re-authentication is enforced when changing roles and is cleared upon power reset.

The module supports concurrent operators but maintains the separation of roles for each. The CO may disable concurrent user sessions by configuring the setting “Disable Concurrent Sessions” on the “GUI Login Authentication” page. If this setting is On, at most one session is allowed per user. Before opening a new session, a user must close the existing session. Note that multiple users may still have concurrent sessions but only one of each. If this setting is Off, due to resource constraints, the module can support up to 500 concurrent TLS sessions.

Table 14 lists the operator roles. The Module does not support a maintenance role. CO is the only role that has access to authentication data after being authenticated.

Table 14. Roles

Name	Type	Operator Type	Authentication Methods
Basic	Role	Owner	Password, Cookie, API Key
Advanced	Role	Owner	Password, Cookie, API Key
Admin	Role	CO	Password, Cookie, API Key
Mesh Node	Role	Other	RF Link

4.3 Approved Services

All services offered by the module and the corresponding SSP access is described in the table below. All services running in the Approved mode are approved services, hence it uses the global indicator ("fips_mode" and "fips_ready" described in Section 2.5) to indicate the mode. Every approved service below has its own indicator, either implicit, or a log message which includes the specific service name (e.g. API Call), the status (success/failure) and the timestamp.

Note, for the sake of brevity, similar SSPs are sometimes collectively represented with an asterisk (*), e.g. DRBG-* represents DRBG-EI, DRBG-State and DRBG-Seed.

Table 15A. Approved Services

Name	Description	Indicator	Input	Output	Security Function Implementation	Roles	Roles SSP Access
Configure Advanced Settings over TLS	Advanced non-security relevant configuration	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (AU, CO-E) Rest of TLS-* (AU, CO-G, E)
Configure Basic Settings over TLS	Basic non-security relevant configuration	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Configure Security Settings over TLS	Security relevant configuration, including uploading license and settings files, and factory reset.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, DRBG-* (CO-E) API-Key (CO-E, I, G, O) Passwords-* (CO-I, E) TLS-Host-Priv, TLS-Host-Pub (CO-G, I, E, O) Rest of TLS-* (CO-G, E) RF-Auth-Key, RF-SK (CO-G, I, O)
Diagnostics Over Serial	Initiation of diagnostics like ping, tcpdump, iperf, etc.	Log message	Serial Shell Command	Command Response	None	Basic, Advanced, Admin	Passwords-* (AU, CO, BU-E)
Diagnostics over TLS	Initiation of diagnostics like iperf, etc.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Reset over TLS	Reset by navigating to Web interface -> Security -> Upgrade -	Implicit – power cycle of module.	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv,

	> Reboot or via API “radio_reset”.						TLS-Host-Pub (CO-E) Rest of TLS-* (CO-G, E)
RF Wireless Link	Establish and transmit data with another module.	Log message showing service start.	None	None	KAS SSC (RF)	Mesh Node	RF-Auth-Key, DRBG-*, RF-SK (MN-E) ECDH-Shared-Secret, ECDH-KDF, RF-DH-Priv, RF-DH-Pub, RF-TDK, RF-TEK (MN-G, E)
Status Over TLS	Retrieve module status through the Web interface.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Update Firmware over TLS	Upload new firmware image through the Web Interface.	Implicit – Auto power cycle of module after completion.	Firmware Image	None	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub, FW-HMAC-Key, (CO-E) Rest of TLS-* (CO-G, E)
Zeroize Over TLS	Zeroization of all CSPs may be achieved either via API (“zeroize”) or by navigating to Web interface -> Tools and Diagnostics -> Factory Reset. through the web interface. Destroy all CSPs.	Implicit – Auto power cycle of module after completion.	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (CO-E, Z) Rest of TLS-* (CO-G, E, Z) RF-*, ECDH-* (Z)
Version Over TLS	Show module version either via API (“build_information”), or by navigating to Web Interface -> Tools and Diagnostics -> Build Information.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Input and Output Data	Input/output of PTT Audio, GPIO, serial and wired network traffic.	None	Input Data	Output Data	None	Unauthenticated	None
Reset	Reset by the reset line or power cycle, or alternatively by connecting to the module over its non-RF interfaces on HTTP port 50000 and send the API command “radio_reset”.	Implicit – Auto power cycle of module after completion.	None	None	None	Unauthenticated	None
Status	Retrieve module status through LEDs, unauthenticated Web.	None	None	LED Status, FIPS	None	Unauthenticated	None

				Error State			
Zeroize	The user may call zeroize without authentication via 1) Attach a USB storage dongle with a “reset license” – a Silvus supplied module- When the module detects this file, it will initiate zeroization. 2) On SC4200 and SC4400, set the MPS position to “Z” and power cycle the module. 3) Connect to the module over its non-RF interfaces on HTTP port 50000 and send the API command “zeroize” followed by “radio_reset”.	1) Implicit – Module will power cycle after zeroize. 2) Implicit – Module LED will blink red when zeroize is completed. 3) Implicit – Module will power cycle after zeroize.	1) Silvus “reset license” file. 2) MPS switch set to “Z” and power cycle 3) “zeroize” API command over HTTP port 50000	1) None 2) None 3) None	None	Unauthenticated	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (Z) Rest of TLS-* (Z) RF-*, ECDH-* (Z)
Wi-Fi Connection	The user may connect to the Wi-Fi AP/Client on the module in authenticated mode (WPA2) or unauthenticated mode (Open). This service uses the non-approved but allowed algorithms AES, PBKDF2 (WPA2). This connection may now be used to access the services listed in Table 15.	None	WPA2 SSID password	Failure OR Success and Network Configuration Response.	None	Unauthenticated	None
VPN Connection	The user may turn on this service to allow peer modules to form a layer-2 connection over a layer-3 network. The user needs to implement their own layer 3 security e.g., IPSEC. Once the service starts, user data may now traverse over this connection to peer modules.	None	None	None	None	Unauthenticated	None

Note: CO=Admin Role, BU=Basic Role, AU=Advanced Role, MN=Mesh Node Role, I=Write, O=Read, E=Execute, G=Generate, Z=Zeroize

Note that none of the above unauthenticated services allow any access to the SSPs other than zeroize, which will destroy them.

4.4 Non-Approved Services

The non-approved mode of operation has the same services running as the approved mode. In addition, the Wireless Link Service allows the use of non-approved/non-allowed DES/ECB and AES/ECB for the encryption/decryption of RF-link packets. The module also allows the use of HTTP instead of TLS, and SNMP, SSH for diagnostics.

Table 15B. Non-Approved Services in Non-Approved Mode

Name	Description	Indicator	Input	Output	Security Function Implementation	Roles	Roles SSP Access
Configure Advanced Settings over HTTP/TLS	Advanced non-security relevant configuration	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (AU, CO-E) Rest of TLS-* (AU, CO-G, E)
Configure Basic Settings over HTTP/TLS	Basic non-security relevant configuration	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Configure Security Settings over HTTP/TLS	Security relevant configuration, including uploading license and settings files, and factory reset.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, DRBG-* (CO-E) API-Key (CO-E, I, G, O) Passwords-* (CO-I, E) TLS-Host-Priv, TLS-Host-Pub (CO-G, I, E, O) Rest of TLS-* (CO-G, E) RF-Auth-Key, RF-SK (CO-G, I, O)
Diagnostics Over Serial	Initiation of diagnostics like ping, tcpdump, iperf, etc.	Log message	Serial Shell Command	Command Response	None	Basic, Advanced, Admin	Passwords-* (AU, CO, BU-E)
Diagnostics over HTTP/TLS	Initiation of diagnostics like iperf, etc.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Diagnostics Over SSH	Initiation of diagnostics like ping, tcpdump, iperf, etc.	Log message	SSH Shell Command	Command Response	None	Basic, Advanced, Admin	Passwords-* (AU, CO, BU-E)
Diagnostics over SNMP	Network monitoring using SNMP.	Log message	SNMP commands	Command Response	None	Unauthenticated	None
Reset over HTTP/TLS	Reset by navigating to Web interface -> Security -> Upgrade -	Implicit – power	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*,

	> Reboot or via API “radio_reset”.	cycle of module.					TLS-Host-Priv, TLS-Host-Pub (CO-E) Rest of TLS-* (CO-G, E)
RF Wireless Link	Establish and transmit data with another module.	Log message showing service start.	None	None	KAS SSC (RF)	Mesh Node	RF-Auth-Key, DRBG-*, RF-SK (MN-E) ECDH-Shared-Secret, ECDH-KDF, RF-DH-Priv, RF-DH-Pub, RF-TDK, RF-TEK (MN-G, E)
RF Wireless Link (Legacy)	Establish and transmit data with another module. This service uses non-approved DES-ECB and AES-ECB. Note that only the LSB 0-63 of the RF-SK are used for DES-ECB.	Log message showing service start.	None	None	None	Mesh Node	RF-SK (MN-E)
Status Over HTTP/TLS	Retrieve module status through the Web interface.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Update Firmware over HTTP/TLS	Upload new firmware image through the Web Interface.	Implicit – Auto power cycle of module after completion.	Firmware Image	None	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub, FW-HMAC-Key, (CO-E) Rest of TLS-* (CO-G, E)
Zeroize Over HTTP/TLS	Zeroization of all CSPs may be achieved either via API (“zeroize”) or by navigating to Web interface -> Tools and Diagnostics -> Factory Reset. through the web interface. Destroy all CSPs.	Implicit – Auto power cycle of module after completion.	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (CO-E, Z) Rest of TLS-* (CO-G, E, Z) RF-*, ECDH-* (Z)
Version Over HTTP/TLS	Show module version either via API (“build_information”), or by navigating to Web Interface -> Tools and Diagnostics -> Build Information.	Log message	API/ Web Input	API/Web Response	KAS SSC (TLS), KTS	Basic, Advanced, Admin	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (BU, AU, CO-E) Rest of TLS-* (BU, AU, CO-G, E)
Input and Output Data	Input/output of PTT Audio, GPIO, serial and wired network traffic.	None	Input Data	Output Data	None	Unauthenticated	None
Reset	Reset by the reset line or power cycle, or alternatively by	Implicit – Auto power	None	None	None	Unauthenticated	None

	connecting to the module over its non-RF interfaces on HTTP port 50000 and send the API command “radio_reset”.	cycle of module after completion.					
Status	Retrieve module status through LEDs, unauthenticated Web.	None	None	LED Status, FIPS Error State	None	Unauthenticated	None
Zeroize	The user may call zeroize without authentication via 1) Attach a USB storage dongle with a “reset license” – a Silvus supplied module-When the module detects this file, it will initiate zeroization. 2) On SC4200 and SC4400, set the MPS position to “Z” and power cycle the module. 3) Connect to the module over its non-RF interfaces on HTTP port 50000 and send the API command “zeroize” followed by “radio_reset”.	1) Implicit – Module will power cycle after zeroize. 2) Implicit – Module LED will blink red when zeroize is completed. 3) Implicit – Module will power cycle after zeroize.	1) Silvus “reset license” file. 2) MPS switch set to “Z” and power cycle 3) “zeroize” API command over HTTP port 50000	1) None 2) None 3) None	None	Unauthenticated	Cookie-Key, API-Key, DRBG-*, Passwords-*, TLS-Host-Priv, TLS-Host-Pub (Z) Rest of TLS-* (Z) RF-*, ECDH-* (Z)
Wi-Fi Connection	The user may connect to the Wi-Fi AP/Client on the module in authenticated mode (WPA2) or unauthenticated mode (Open). This service uses the non-approved but allowed algorithms AES, PBKDF2 (WPA2). This connection may now be used to access the services listed in Table 15.	None	WPA2 SSID password	Failure OR Success and Network Configuration Response.	None	Unauthenticated	None
VPN Connection	The user may turn on this service to allow peer modules to form a layer-2 connection over a layer-3 network. The user needs to implement their own layer 3 security e.g., IPSEC. Once the service starts, user data may now traverse over this	None	None	None	None	Unauthenticated	None

	connection to peer modules.						

4.5 External Software/Firmware Loaded

The module firmware may be updated by navigating to the Web GUI -> Tools and Diagnostics -> Firmware Upgrade page and uploading the Silvus provided firmware package. The module will first shut down all interfaces that allow data output before beginning this service. The module will then authenticate and validate the package, update the firmware and auto-power cycle. Once the module powers back up, it is executing the new firmware image.

The firmware package has a manifest file and the actual firmware image. The firmware image is an encrypted binary blob containing the firmware files. The module performs two checks on the externally loaded firmware.:

- HMAC-SHA-256 verification of the firmware file by comparing it to the checksum inside the manifest.

If there is any error during firmware load, the module will run a self-test and upon success, will re-enable the interfaces and bring itself back online. These errors are logged, and the CO may download the log file to check the reason for the upgrade failure. If the self-test fails, the module will go into the FIPS error state. If the upgrade succeeds, the module will power cycle.

4.7 Cryptographic Output Actions and Status

All services other than the RF Wireless Link service are user initiated. The RF Wireless Link service will automatically start outputting encrypted user data over the RF interface. This service is enabled when the module is configured into the FIPS Approved Mode.

To enable this service, the following two independent actions are required. CO will first enable Encryption for the Wireless Link Service, the module will verify this and then the module will check if all other Approved mode requirements are completed before turning on the wireless interface. At this point, the module is fully in the Approved mode and will begin encrypting and decrypting data on the wireless interface.

The FIPS status indicator being “green” confirms that the self-initiated cryptographic output service is currently enabled and working.

4.8 Additional Information

Authentication Response
Password authentication with invalid passwords or HMAC signatures would be responded by “Invalid Authentication”. RF Link Encryption service will drop any KAS packets that are not correctly signed with the RF-Auth-Key

5.0 Software/Firmware security

5.1 Integrity Techniques

The module, upon power up will run a bootloader - U-Boot. U-Boot performs a CRC-32 verification of the firmware image before eventually handing control to the OS.

If the firmware integrity test fails, U-Boot performs a CRC-32 check on a secondary copy of the firmware image. eventually handing control to the OS if the check passes. If this firmware also fails integrity checks, the module will not power up. The LED changes to red and the module is considered inoperable and will need factory repair.

5.2 Initiate on Demand

Integrity checks are initiated on demand by power cycling the module.

6.0 Operational environment

6.1 Operational Environment Type and Requirements

Type of Operating Environment: Limited

7.0 Physical security

The cryptographic module includes the following physical security mechanisms:
- Production-grade components and production-grade opaque enclosure
- Two (2) tamper-evident seals applied during manufacturing, please see Figures 2 (SC4200), 6,7 (SC4400), 11 (SL4200), 14 (SM4200).

7.1 Mechanisms and Actions Required

Tamper evident seals are applied at Manufacturing.

Table 16. Mechanisms and Actions Required

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Tamper evident seals.	Every 12 months.	If the module shows any signs of tamper, the CO should zeroize the module and contact the manufacturer.

8.0 Non-invasive security

N/A

9.0 Sensitive security parameters management

9.1 Storage Areas

The module stores non-ephemeral SSPs in a partition on a NAND chip. Ephemeral SSPs are stored in system memory (RAM). The module also has an e-fuse EEPROM to store the Secure boot SSP used for power-up firmware integrity testing. Note that this is not claimed for security.

Table 17. Storage Areas

Name	Description	Persistence Type
RAM	Volatile system memory.	Dynamic
NAND	Non-volatile system memory.	Static
EFUSE	Non-volatile e-fuse memory.	Static

9.2 SSP Input-Output Methods

Table 18. SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Input	User, App	Module	Encrypted	Automated	Electronic	AES-GCM
API Output	Module	User, App	Encrypted	Automated	Electronic	AES-GCM
Pre-Loaded	Manufacturer	Module	Plaintext	N/A	N/A	N/A
RF KAS Output	Module	Peer Module	Plaintext (Public Key)	Automated	Electronic	KAS-ECC-SSC 56Ar3
TLS KAS Output	Module	TLS Client	Plaintext (Public Key)	Automated	Electronic	KAS-ECC-SSC 56Ar3

The module does not support manual SSP entry or intermediate key generation output. Plaintext output of CSPs is only allowed over a secure channel (TLS v1.3).

9.3 SSP Zeroization Methods

Table 19. SSP Zeroization Methods

Method	Description	Rationale	Operator Initiation Capability
API Zeroize	The zeroization is performed by the module overwriting zeroes to the memory location occupied by the SSP and further deallocating that area. The completion of API Zeroize will cause the module to auto power-cycle.		1. Zeroize over TLS on the Web GUI or by calling the API “zeroize”. 2. Zeroize by moving MPS to position “Z” and power-cycling the module.

			3. Attach a USB storage dongle with the Silvus provided “reset license”. 4. Zeroize by calling the API “zeroize” on HTTP port 50000 over the local (non-RF) interfaces.
Memory Cleanse	Zeroize for temporary and ephemeral SSPs by overwriting with zeroes to the memory location.		N/A. Not initiated by operator.

9.4 SSPs

Table 20. SSPs - Part 1

Name	Description	Size	Strength	Type	Generated By	Established By
Cookie-Key	Used to sign authentication cookies using HMAC-SHA-384	256 bits	256 bits	HMAC-SHA-384 Key	DRBG (Internal)	N/A
API-Key	Used to verify API requests	256 bits	256 bits	HMAC-SHA-256	DRBG (Internal) or see input	N/A
FW-HMAC-Key	Used to authenticate firmware load file.	256 bits	256 bits	HMAC-SHA-256	N/A	N/A
DRBG-EI	DRBG entropy input.	N/A	256 bits	AES CTR DRBG with 256 bits of security.	Silvus Clock Jitter Entropy Module (Internal)	N/A
DRBG-Seed	DRBG Seed	256 bits key, 128 bits IV.	256 bits	AES CTR input seed (key, IV)	DRBG (Internal)	N/A

DRBG-State	Internal DRBG State – K & V.	256 bits K, 128 bits V	256 bits	AES CTR DRBG internal state.	DRBG (Internal)	N/A
Passwords	Web authentication passwords.	Minimum 8 characters.	N/A	Alpha-Numeric Passwords: Allowed Special Characters @#!~$%^&*()-+/:.,<>?|_[]="'\{};`	N/A	N/A
Passwords-HASH	SHA-384 hashes of the passwords.	384 bits	192 bits	SHA-384	SHS (Internal)	N/A
RF-DH-Priv	P-521 ECDHE ephemeral private key.	521 bits	256 bits	SP-800 56Ar3 compliant P-521 private key.	ECDSA KeyGen (Internal)	N/A
RF-Auth-Key	HMAC-SHA-256 key used to authenticate RF Wireless Link KAS messages.	256 bits	256 bits	HMAC-SHA 256	DRBG (Internal) or see input	N/A
ECDH-Shared-Secret	P-521 Shared secret established after successful 56A KAS for RF Wireless Link.	521 bits	256 bits	P-521 primitive Z computation.	N/A	KAS-ECC-SSC 56Ar3
ECDH-KDF	56Cr1 compliant KDF to generate RF-TDK and RF-TEK for RF Wireless Link.	256 bits	256 bits	56Cr1 One-Step KDF	KDA One Step SP-800 56Cr1 (Internal)	N/A
RF-TDK	AES-GCM 256 bit key generated after ECDH-KDF for decrypting data on RF	256 bits	256 bits	AES GCM 256	KDA One Step SP-800 56Cr1 (Internal)	N/A

	Wireless Link.					
RF-TEK	AES-GCM 256 bit key generated after ECDH-KDF for encrypting data on RF Wireless Link.	256 bits	256 bits	AES GCM 256	KDA One Step SP-800 56Cr1 (Internal)	N/A
RF-SK	AES-GCM 256 bit key used to encrypt and decrypt data on RF Wireless Link.	256 bits	256 bits	AES GCM 256	DRBG (Internal) or see input	N/A
TLS-DH-Priv	TLS ECDH (ephemeral) private key.	256, 384, 521 bits.	128, 192, 256 bits.	P-256, P-384, P-521 private key.	ECDSA KeyGen (Internal)	N/A
TLS-Host-Priv	TLS server host certificate private key.	256, 384, 521 bits	128, 192, 256 bits.	P-256, P-384, P-521 private key.	N/A	N/A
TLS-PMS	TLS pre-master secret.	256, 384, 521 bits.	128, 192, 256 bits.	P-256, P-384, P-521 primitive Z computation.	N/A	KAS-ECC-SSC 56Ar3
TLS-MS	TLS master-secret.	384 bits.	256 bits	TLS PRF computation.	N/A	KDA HKDF
TLS-KDF	56Cr1 compliant KDA HKDF to generate TLS-MS, TLS-SENC, TLS-SMAC.	256 bits	256 bits	56Cr1 KDA HKDF	KDA Two Step SP-800 56Cr1 (Internal)	N/A
TLS-SENC	TLS session encryption key.	256 bits.	256 bits.	AES GCM 256 bit key.	N/A	KDA HKDF
TLS-SMAC	TLS session authentication key.	256 bits.	256 bits.	HMAC-SHA-384 256 bit key.	N/A	KDA HKDF
RF-DH-Pub	P-521 ECDH ephemeral public key.	521 bits	256 bits	SP-800 56A compliant P-521 private key.	ECDSA KeyGen (Internal)	N/A

TLS-Host-Pub	TLS server host certificate public key.	256, 384, 521 bits	128, 192, 256 bits	P-256, P-384, P-521 public key.	ECDSA Keygen (Internal) Or see input	N/A
TLS-DH-Pub	TLS ECDH (ephemeral) public key.	256, 384, 521 bits.	128, 192, 256 bits	P-256, P-384, P-521 public key.	ECDSA KeyGen (Internal)	N/A

Table 21. SSPs – Part 2

Name	Used By	Inputs/Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
Cookie-Key	HMAC, SHS	N/A	RAM	Duration of power to module. Auto re-generated on every power cycle.	API Zeroize	CSP	Passwords, Passwords-HASH
API-Key	HMAC, SHS	API Input, API Output	NAND	N/A	API Zeroize	CSP	None
FW-HMAC-Key	HMAC, SHS	Pre-Loaded	NAND	N/A	N/A	CSP	None
DRBG-EI	DRBG	N/A	RAM	DRBG instantiation < 1s.	API Zeroize, Memory Cleanse	CSP	DRBG-Seed, DRBG-State
DRBG-Seed	DRBG	N/A	RAM	DRBG instantiation < 1s.	API Zeroize, Memory Cleanse	CSP	DRBG-EI, DRBG-State
DRBG-State	DRBG	N/A	RAM	Duration of power to module.	API Zeroize, Memory Cleanse	CSP	DRBG-EI, DRBG-Seed
Passwords		API Input	RAM	Password Authentication < 1s.	API Zeroize	CSP	Passwords-HASH
Passwords-HASH	SHS	N/A	NAND	N/A	API Zeroize	CSP	Passwords
RF-DH-Priv	KAS-ECC-SSC	N/A	RAM	RF Link KAS handshake < 1s.	API Zeroize, Memory Cleanse	CSP	RF-DH-Pub, ECDH-Shared-Secret, ECDH-KDF, RF-

							TDK, RF-TEK
RF-Auth-Key	HMAC, SHS	API Input, API Output	NAND	N/A	API Zeroize	CSP	RF-DH-Pub
ECDH-Shared-Secret	KDA One Step	N/A	RAM	RF Link KAS handshake < 1s.	API Zeroize, Memory Cleanse	CSP	RF-DH-Pub, RF-DH-Priv, ECDH-KDF, RF-TEK, RF-TDK
ECDH-KDF	KDA One Step	N/A	RAM	RF Link KAS handshake < 1s.	API Zeroize, Memory Cleanse	CSP	RF-DH-Pub, RF-DH-Priv, ECDH-Shared-Secret, RF-TDK, RF-TEK
RF-TDK	AES GCM	N/A	RAM	Duration of power to module.	API Zeroize, Memory Cleanse	CSP	RF-DH-Pub, RF-DH-Priv, ECDH-Shared-Secret, ECDH-KDF, RF-TEK
RF-TEK	AES GCM	N/A	RAM	Duration of power to module.	API Zeroize, Memory Cleanse	CSP	RF-DH-Pub, RF-DH-Priv, ECDH-Shared-Secret, ECDH-KDF, RF-TDK
RF-SK	AES GCM	API Input, API Output	NAND	N/A	API Zeroize	CSP	None
TLS-DH-Priv	KAS-ECC-SSC	N/A	RAM	TLS Handshake < 1s.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-PMS, TLS-MS,

							TLS-KDF, TLS-SENC, TLS-SMAC
TLS-Host-Priv	ECDSA SigGen, SigVer	API Input, API Output	NAND	N/A	API Zeroize	CSP	TLS-Host-Pub
TLS-PMS	KDA HKDF 56Cr1	N/A	RAM	TLS Handshake < 1s.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-DH-Priv, TLS-MS, TLS-SENC, TLS-SMAC
TLS-MS	KDA HKDF 56Cr1	N/A	RAM	TLS Handshake < 1s.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-DH-Priv, TLS-PMS, TLS-SENC, TLS-SMAC, TLS-KDF
TLS-KDF	KDA HKDF 56Cr1	N/A	RAM	TLS Handshake < 1s.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-DH-Priv, TLS-PMS, TLS-MS, TLS-SENC, TLS-SMAC
TLS-SENC	AES GCM	N/A	RAM	TLS session duration.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-DH-Priv, TLS-PMS, TLS-

							MS, TLS-KDF, TLS-SMAC
TLS-SMAC	HMAC, SHS	N/A	RAM	TLS session duration.	API Zeroize, Memory Cleanse	CSP	TLS-DH-Pub, TLS-DH-Priv, TLS-PMS, TLS-MS, TLS-KDF, TLS-SENC
RF-DH-Pub	KAS-ECC-SSC	RF KAS Output	RAM	RF Link KAS handshake < 1s.	API Zeroize, Memory Cleanse	PSP	RF-DH-Priv, ECDH-Shared-Secret, ECDH-KDF, RF-TDK, RF-TEK
TLS-Host-Pub	ECDSA SigGen, SigVer	API Input, API Output	NAND	N/A	API Zeroize	PSP	TLS-Host-Priv
TLS-DH-Pub	KAS-ECC-SSC	TLS KAS Output	RAM	TLS Handshake < 1s.	API Zeroize, Memory Cleanse	PSP	TLS-DH-Priv, TLS-PMS, TLS-MS, TLS-SENC, TLS-SMAC

Note, there is a configuration API (enc_key_volatile_disable") available to the CO that when set to "0" would force the storage of certain SSPs in RAM instead of NAND. When the module power cycles, these SSPs would revert to their factory default values, hence requiring a FIPS re-configuration after every power cycle. The following are the affected SSPs:

1. RF-Auth-Key
2. API-Key
3. RF-SK

By default, this API is set to "1" and these SSPs are stored in NAND.

10.0 Self-tests

Each time the Module is powered up, it tests that the cryptographic algorithms still operate correctly, and that sensitive data have not been damaged. Power up self–tests are available on demand by power cycling the module.

On power up or reset, the Module performs the pre-operational self-tests described in Table 22 followed by the power-up conditional tests in Table 23 below without any operator action required. All data output via the data output interface is inhibited when an error state exists and during self-tests. All KATs must be completed successfully prior to any other use of cryptography by the Module.

If any of the KATs fails, the Module enters a fatal error state. The web page will switch to using HTTP only vs. HTTP/TLS and will only display a single page showing more information about the error state, e.g., which KAT failed. The user may then reboot the module to resume operation. If all KATs are completed successfully, the module will enter a functional state. The user may navigate to the Web interface and look at the informational side-panel. The entry “FIPS Status” should show a Green-filled circle which indicates the module is in Approved mode. If it is Amber, it indicates that the module is in a transitional FIPS state and requires manual configuration by the CO to complete the transition. If it is Red, it indicates the module is in Non-Approved mode.

The module will go to the same FIPS error state in case of errors during all non-power-up conditional tests except the firmware load test. If the firmware load test fails, the module will go into a temporary soft error state and run a self test. If the self test fails, the module will go into the FIPS error state. If it passes, the module will log the reason for the original failure and bring the module back into operable state. The CO may download the log file to find the reason for failure and/or retry. The module does not have any status output except for LEDs, so during the FIPS error state the Ethernet interface will still be up and will allow connecting to the HTTP error page.

The following are the ways to power cycle the radio and run the power-up self-tests:

1. Authenticate as CO and Invoke the “Reset over TLS” service either through the GUI (Tools and Diagnostics -> Upgrade Firmware -> Reboot) or by calling the API “radio_reset”.
2. Invoke the unauthenticated reset service by toggling the MPS/Power Switch or calling the API “radio_reset” on HTTP port 50000 through a non-RF interface.

10.1 Pre-Operational Self-Tests

Table 22. Pre-Operational Self-Tests

Algorithm	Implementation	Test Properties	Test Method	Type	Indicator	Details
CRC-32	CRC-32 checksum comparison	CRC-32 checksum comparison	CRC-32 checksum comparison.	FW Integrity	Upon success, the	The U-Boot bootloader

					module will power up and the LED will change from RED to GREEN. If the integrity test fails, the module will try a fallback image. If that fails too, the LED will stay RED, and the module will not power up (bricked).	will check the CRC-32 checksum of the firmware image against the value stored in the firmware file header.

10.2 Conditional Self-Tests

Table 23. Conditional Self-Tests

Concerning the indicator, for all conditional tests with the power-up condition below, successful completion would cause the module to finish powering up, with the associated indicators for Approved/Non-Approved mode. A failure would put the module in FIPS error mode.

Success will be given in the API log. Failure will result in a hard error state.

Algorithm	Implementation	Test Properties	Test Method	Type	Indicator	Details	Condition
DRBG	A3422	SP-800 90A Health Tests	Health Check	Health Check	Listed above table (1)		Power-Up
ENT (NP)	E25	SP-800 90B Health Tests	Repetition Count Test, Stuck Test, Adaptive	Health Check	Listed above table (1)	Health Check	Power-Up

			Proportion Test, Lag Predictor Test.				
AES CTR	A3422	256-bit	KAT	CAST	Listed above table (1)	Encrypt/Decrypt	Power-Up
AES GCM	A3422	256-bit	KAT	CAST	Listed above table (1)	Encrypt/Decrypt	Power-Up
AES ECB	A3422	256-bit	KAT	CAST	Listed above table (1)	Encrypt/Decrypt	Power-Up
SHA2	A3422	1, 256, 384, 512	KAT	CAST	Listed above table (1)	Hash	Power-Up
SHA3	A3425	256	KAT	CAST	Listed above table (1)	Hash	Power-Up
HMAC-SHA	A3422	1, 256, 384	KAT	CAST	Listed above table (1)	MAC	Power-Up
KAS-SSC	A3422	P-521	KAT	CAST	Listed above table (1)	Z computation	Power-Up
ECDSA	A3422	P-521, SHA-256	SigGen, SigVer KAT	CAST	Listed above table (1)	Sign and Verify	Power-Up
DRBG	A3422	AES CTR-DRBG	KAT	CAST	Listed above table (1)	RBG	Power-Up
TLS HKDF	A3422	V1.3	KAT	CAST	Listed above table (1)	KDF	Power-Up
56C KDF	A3422	1-Step SHA-256	KAT	CAST	Listed above table (1)	KDF	Power-Up
AES ECB	A3420 or A3421	256-bit	KAT	CAST	Listed above	Encrypt	Power-Up

					table (1)		
AES GCM	A3420 or A3421	256-bit	KAT	CAST	Listed above table (1)	Encrypt/Decrypt	Power-Up
DRBG	A3422	SP-800 90A Health Tests	Health Check	Health Check	Listed above table (2)	Health Check	Performed when a random value is requested from the entropy source as per SP 800-90B.
ECDSA	A3422	P-256 P-384 P-521	PCT	PCT	Listed above table (2)	Signature Generation, Signature Verification, Key Verification	Key Generation
Firmware Load	A3422	HMAC-SHA-256	Compare hash results	Firmware Load	Listed above table (2)	MAC Verification	Performed upon a firmware load request.
Entropy Source	E25	SP-800 90B Health Tests	Repetition Count Test, Stuck Test, Adaptive Proportion Test, Lag Predictor Test.	Health Check	Listed above table (2)	Health Check	Performed when a random value is requested from the entropy source as per SP 800-90B.
SP 800-56Ar3 Assurances	A3422	SP 800-56Ar3 Assurances	PCT	PCT	Listed above table (2)	Public Key Validation Private Key Validation ECCDH PCT: Key Pair Computation	Performed conditionally per Section 5.5.2, 5.6.2 and/or 5.6.3.

10.3 Periodic Self-Tests

On demand self-tests can be invoked by powering cycling the module.

10.4 Error States

Table 24. Error States

State Name	Description	Conditions	Recovery Method	Indicator
FIPS Error State	Catch-all error state for unrecoverable errors.	Any failure in: 1) Conditional tests occurring on power-up. 2) SP-800 90B health checks. 3) SP-800 56A assurances. 4) ECDSA PCT 5) Self tests run due to a failure of the firmware load test.	Power cycle.	Web GUI will change from TLS to HTTP with a single page indicating that a FIPS error has occurred.
Soft Error State	Catch-all error state for recoverable errors.	Firmware Load Test	Retry.	Module will temporarily shut down all interfaces and run a self test. If the self test succeeds, the interfaces will be brought up again, and the CO may download the log file to check the reason for the failure.

10.5 Operator Initiation

Initiation operations are listed in section 2.5 of this Security Policy.

10.6 Additional Information

N/A

11.0 Life-cycle assurance

11.1 Startup Procedures

The module is shipped ready to use in the FIPS Non-Approved mode. The CO needs to follow the instructions in Section 2.5 to configure the module into the FIPS Approved mode.

11.2 Administrator Guidance

Please refer to Section 2.5 for the administrator guidance for configuring the module into the FIPS Approved mode.

11.3 Non-Administrator Guidance

Please refer to the Silvus User Manual for non-security related usage information.

11.4 Maintenance Requirements

N/A

11.5 End of Life

The operator is required to use the “Zeroize over TLS” OR the “Zeroize” service to zeroize all the keys when the module reaches end of life.

11.6 Additional Information

N/A

12.0 Mitigation of other attacks

No mitigation of other attacks is implemented by the module.