EF Johnson Technologies
Johnson Encryption Machine 2 (JEM2)

FIPS 140-2 Cryptographic Module
Non-Proprietary Security Policy

Hardware Versions: R035-3900-180-00 and R035-3900-280-01

Firmware Version: 4.2

Date: 24 September 2021

Table of Contents

List of Tables

List of Figures

1 Introduction

This document defines the Security Policy for the EF Johnson Technologies Johnson Encryption Machine 2 (JEM2) module, hereafter denoted the Module. The Module is a PCIe-based multi-chip embedded cryptographic module. The primary purpose for this device is to provide cryptographic services to a host computer for the purpose of supporting Project 25 (P25) infrastructure equipment such as dispatch consoles and key management facilities. The Module includes those services necessary to support FIPS-140-2 validated encryption and decryption of P25 voice and data and key management and storage. The Module also includes non-validated legacy services to support P25 DES encryption operation.

The Module meets FIPS 140-2 overall Level 1 requirements. The security rules specified in this document include rules derived from the FIPS 140-2 standard as well as requirements imposed by EF Johnson Technologies.

The JEM2 crypto module (Hardware Versions R035-3900-180-00 and R035-3900-280-01, Firmware Version 4.2) is a multi-chip embedded module consisting of several integrated circuits and additional discrete components that are assembled onto a printed circuit board. The operator can determine the correct firmware version by reading the DCB and confirming the correct version. The operator can determine the correct hardware version by reading the printed version from the board.

Table 1 – Cryptographic Module Configurations

	Module	HW P/N and Version	FW Version	OE (if applicable)
1	JEM2 Full-Height	R035-3900-180-00	4.2	N/A
2	JEM2 Half-Height	R035-3900-280-01	4.2	N/A

The Module is intended for use by US Federal agencies and other markets that require FIPS 140-2 validated P25 cryptographic services. The Module is a multi-chip embedded embodiment. The cryptographic boundary of the JEM2 encompasses the JEM2 PCB and all the hardware, software and firmware components contained therein. No components within the confines of the JEM2 PCB are excluded from the cryptographic boundary. Note that all cryptographic keys stored in the JEM2 key store are contained within the cryptographic boundary.

The FIPS 140-2 security levels for the Module are as follows:

Table 2 – Security Level of Security Requirements

Security Requirement	Security Level
Cryptographic Module Specification	Level 1
Cryptographic Module Ports and Interfaces	Level 1
Roles, Services, and Authentication	Level 1
Finite State Model	Level 1
Physical Security	Level 1
Operational Environment	N/A

Security Requirement	Security Level
Cryptographic Key Management	Level 1
EMI/EMC	Level 1
Self-Tests	Level 1
Design Assurance	Level 1
Mitigation of Other Attacks	N/A

The Module implementation is compliant with:

- FIPS 140-2 Security Requirements for Cryptographic Modules
- Implementation Guidance for FIPS 140-2 and the CMVP 8/28/20
- FIPS 180-4 Secure Hash Standard (SHS)
- FIPS 186-4 Digital Signature Standard (DSS)
- FIPS 197 Advanced Encryption Standard
- FIPS 198-1 Keyed-Hash Message Authentication Code (HMAC)
- SP 800-38A Recommendation for Block Cipher Modes of Operation – Methods and Techniques
- SP 800-57 Recommendation for Key Management
- SP 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators
- SP 800-38F Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
- TIA-102.AAAB-A Security Services Overview
- TIA-102.AAAD Block Encryption Protocol
- TIA-102.AACA-1 Key Management Security Requirements
- TIA-102.AACE-A Link Layer Authentication

Hardware and Physical Cryptographic Boundary

The physical form of the Module is depicted in Figures 1 and 2; the red outline depicts the physical cryptographic boundary.

The cryptographic boundary of the JEM2 encompasses the JEM2 PCB and all the hardware, software and firmware components contained therein. No components within the confines of the JEM2 PCB are excluded from the cryptographic boundary. Note that all cryptographic keys stored in the JEM2 key store are contained within the cryptographic boundary.

The Module is a PCIe card that is designed to be installed into a compatible host computer. The module receives its power from the computer's PCIe interface port. The module communicates with the host computer through the PCIe interface. Processing is done on a digital signal processor (DSP). The module contains both volatile and non-volatile memory. Non-volatile memory stores the executable code of the DSP and the cryptographic keys of the JEM2 key store. The module will transfer both the executable code and the contents of the key store from non-volatile to volatile memory after power-up. Status output is provided via LEDs. A hardware NDRNG circuit provides random data.

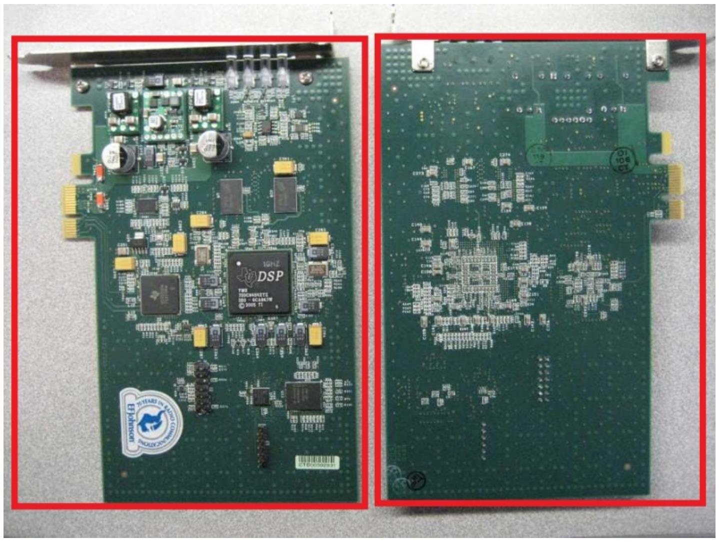

Figure 1 – Module (R035-3900-180-00)

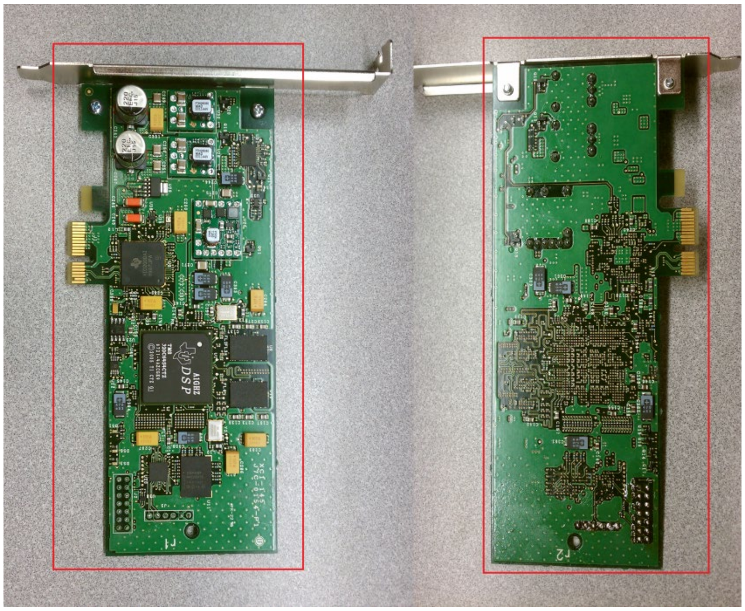

Figure 2 – Module (R035-3900-280-01)

The Module incorporates four physical ports and five logical interfaces. The physical ports are the PCIe connection to the host computer and the power indicator LED, FIPS mode LED, and status LED on the back of the JEM2 PCB. The five logical interfaces include the data input interface, data output interface, control input interface, status output interface, and power interface. The following table shows which logical interface each physical port supports.

Table 3 – Ports and Interfaces

Port	Logical Interface Type
PCIe	Power, Control in, Data in, Data out, Status out
Power Indicator LED	Status out
FIPS Mode LED	Status out
Status LED	Status out

Data Input Interface

The data input interface transfers data over the PCIe port from the host to the module. The data input interface is disabled during self-test and error states.

Data Output Interface

The data output interface transfers data over the PCIe port from the module to the host. The data output interface is disabled during self-test and error states.

Control Input Interface

The control input interface transfers control data over the PCIe port from the host to the module. Control input is achieved through direct buffer commands and messaging header information.

Status Output Interface

The status output interface transfers status data over the PCIe port from the module to the host and sets physical indicator LEDs. Status output is achieved over the PCIe port through direct buffer status reports and messaging status codes. The power indicator LED indicates the device has power. The FIPS mode LED indicates whether the device is currently in FIPS approved or FIPS non-approved mode. The FIPS mode LED is green when the device is in FIPS approved mode and unlit when the device is in FIPS non-approved mode. The status LED indicates whether the device is currently running normally or is in an error state. The status LED is green when the device is in normal operation and red when the device is in an error state.

Power Interface

The power interface consists of the +12 V and +3.3 V pins on the PCIe port. These pins supply power to voltage regulator circuitry on the module PCB that creates additional voltages needed by the active components of the module. Reset circuitry holds the DSP in reset during under-voltage conditions.

1.1 Firmware and Logical Cryptographic Boundary

Figure 3 depicts the Module operational environment.

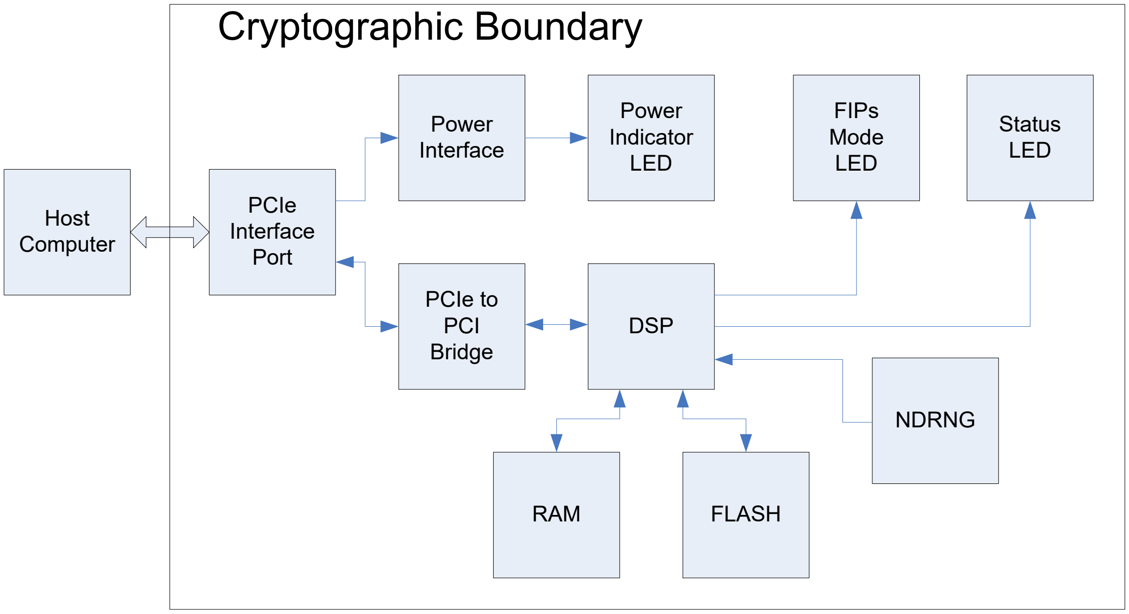

Figure 3 – Module Block Diagram

1.2 Modes of Operation

The module supports a FIPS-140-2 approved operational mode and a non-approved operational mode. Each mode offers cryptographic services to support P25 infrastructure security requirements. The host can set the desired operational mode of the module by issuing a *Continue FIPS* or *Continue NonFIPS* command while the module device is in a stopped condition. CSPs are not shared between the approved and non-approved modes; when transitioning between modes of operation keys are zeroized.

To verify that a module is in the Approved mode of operation, check the FIPS Mode LED. If the FIPS Mode LED is on (green) the Module is operating in the Approved mode.

FIPS-140-2 Approved Operational Mode Security Functions

Approved Security Functions

1. AES-256 ECB
2. AES-256 CBC
3. AES-256 OFB
4. AES-192 ECB
5. AES-192 CBC
6. AES-192 OFB
7. AES-128 ECB
8. AES-128 CBC
9. AES-128 OFB
10. ECDSA Signature Verification: Curve P-256
11. SHA-1
12. SHA-256
13. SHA-512
14. HMAC: SHA-256
15. DRBG: SHA-512 Hash DRBG (SP 800-90A)
16. CKG
17. AES Key Wrap (SP 800-38F)

Non-Approved Security Functions (allowed for use in FIPS Approved Mode)

1. AES MAC (vendor affirmed; P25 AES OTAR)
2. NDRNG (used to generate seed values for the Approved DRBG)

Non-Approved Operational Mode Security Functions

1. DES ECB
2. DES CBC
3. DES OFB

Note: All security functions available in the FIPS-140-2 approved mode are also available in the non-approved mode.

2 Cryptographic Functionality

The Module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in the tables below.

Table 4 – Approved and CAVP Validated Cryptographic Functions

Algorithm	Description	Cert #
AES	[FIPS 197, SP 800-38A] Functions: Encryption, Decryption Modes: ECB, CBC, OFB Key sizes: 128, 192, 256 bits	#3436
AES Key Wrap/Unwrap	[SP 800-38F] Functions: Wrap, Unwrap Modes: AES-KW Key sizes: 128, 192, 256 bits	#3437
CKG	[SP 800-133] Sections 4 and 6.1	Vendor Affirmed
DRBG	[SP 800-90A] Functions: Hash DRBG Security Strengths: 256 bits	#837
ECDSA	[FIPS 186-4] Functions: Signature Verification Curves/Key sizes: P-256	#692
HMAC	[FIPS 198-1] Functions: Generation, Verification SHA sizes: SHA-256	#2187
KTS	[SP 800-38F] Functions: Wrap, Unwrap Modes: AES-KW Key sizes: 128, 192, 256 bits	#3437
SHS	[FIPS 180-4] Functions: Digital Signature Verification, non-Digital Signature Applications SHA sizes: SHA-1, SHA-256, SHA-512	#2838

Table 5 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Description
AES OTAR	AES MAC (AES Cert. #3436, vendor affirmed; P25 AES OTAR)
NDRNG	[Annex C] Hardware Non-Deterministic RNG. The NDRNG output is used to seed the FIPS Approved DRBG with 200 bytes with a minimum of 6.746 bits of min-entropy per output byte, which is sufficient to provide the full 256-bits of security strength.

Non-Approved Cryptographic Functions for use in non-FIPS mode only:

- DES

2.1 Critical Security Parameters

All CSPs used by the Module are described in this section. All usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in Section 4.

Table 6 – Critical Security Parameters (CSPs)

CSP	Description / Usage
AES Traffic Encryption Keys (TEK)	This is an AES OFB key used to encrypt voice and data. Possible TEK sizes are 128-bit, 192-bit, and 256-bit. These keys are key wrapped by the KSK.
AES Key Encryption Key (KEK)	This is an AES-KW key used to encrypt TEKs and KEKs during the key management operations. Possible KEK sizes are 128-bit, 192-bit, and 256-bit. The KEK size is restricted by the module to be greater than or equal to the size of the key to be encrypted. These keys are key wrapped by the KSK.
Key Storage Key (KSK)	This is an AES-KW key used to key encrypt all keys for storage. This key is 256 bits.
Key MAC Key (KMACK)	This is used to compute the HMAC for a stored key. The KMACK is 256-bit.
DRBG Working State	The DRBG working state is used by the DRBG security function of the Module to generate the KSK, KMACK, and TEK keys. The seed and nonce used to generate pseudo-random numbers is obtained at the same time by sampling a hardware NDRNG circuit on the Module.
P25 System Authentication Key (P25SAK)	A 128-bit AES ECB key used for TIA P25 Phase 2 System Authentication.
P25 System Session Authentication Key (P25SSAK)	A 128-bit AES ECB key generated from P25SAK and used for TIA P25 Phase 2 System Authentication.
P25 System Mutual Session Authentication Key (P25SMSAK)	A 128-bit AES ECB key generated from P25SAK and used for TIA P25 Phase 2 System Authentication.

2.2 Public Keys

Table 7 – Public Keys

Key	Description / Usage
ECDSA (P-256) Signature Verification public key	This public key is used to verify the signature of the downloaded firmware.

3 Roles, Authentication and Services

3.1 Assumption of Roles

The module supports two distinct operator roles, User and Cryptographic Officer (CO). The cryptographic module enforces the separation of roles by restricting one role per module reset. The User and CO roles are mutually exclusive and cannot exist concurrently.

Table 8 lists all operator roles supported by the module. The Module does not support a maintenance role. The Module does not support concurrent operators. The Module does not support authentication for either the User or CO roles.

Table 8 – Roles Description

Role ID	Role Description
CO	Cryptographic Officer – This role is implicitly assumed when an operator uses one of the CO services. See Table 9 below.
User	User – This role is implicitly assumed when the operator uses one of the User services. See Table 9 below.

3.2 Services

The Module provides services to the host using a request/response based proprietary messaging protocol. The host will make a service request and the Module will perform the service and issue a response to the host. For infrastructure operations that require multiple service calls for completion of the operation, a single stream ID is used for that operation. The Module associates the stream ID with a particular key, algorithm state information, and other parameters. The Module provides the services that are described in detail in the sections that follow and are summarized in the table included below.

Note: All the services below, except Code Update, are also available in the non-approved mode of operation. Code Update is only available in approved mode of operation. In addition to the approved mode of operation services, the non-approved mode of operation offers the same services that require an AES key but with a DES key.

All services implemented by the Module are listed in the table below. Each service description also describes all usage of CSPs by the service.

Table 9 – Services

Service Name	Algorithms Used	Key Parameters	Inputs & Outputs				Role		CSP Access
			Control Input	Data Input	Data Output	Status Output	User	CO	
Load Key	None	256 AES 192 AES 128 AES	Header Info	Key Index	None	Success/ Fail	X		Associate Stored Key with Stream ID Wrap with KSK
Validate Key	None	256 AES 192 AES	Header Info	Key Index	None	Success/ Fail	X		Verify Key Exists

		128 AES							
Get Random Data	DRBG	None	Header Info	None	DRBG output data	Success/ Fail	X		Access DRBG
Enter Voice Tx	DRBG	64 IV	Header Info	None	MI data	Success/ Fail	X		Generate IV Access DRBG
Encrypt LC	AES OFB	256 AES	Header Info	Plaintext LC data	Ciphertext LC data	Success/ Fail	X		Encrypt with TEK
Encrypt Voice Frame	AES OFB	256 AES	Header Info	Plaintext voice data	MI data Ciphertext voice data	Success/ Fail	X		Encrypt with TEK
Encrypt LSD	AES OFB	256 AES	Header Info	Plaintext LSD data	Ciphertext LSD data	Success/ Fail	X		Encrypt with TEK
Enter Voice Rx	None	64 IV	Header Info	MI data	None	Success/ Fail	X		None
Decrypt LC	AES OFB	256 AES	Header Info	Ciphertext LC data	Plaintext LC data	Success/ Fail	X		Decrypt with TEK
Decrypt Voice Frame	AES OFB	256 AES	Header Info	MI data Ciphertext voice data	Plaintext Voice data	Success/ Fail	X		Decrypt with TEK
Decrypt LSD	AES OFB	256 AES	Header Info	Ciphertext LSD data	Plaintext LSD data	Success/ Fail	X		Decrypt with TEK
Set MI	None	64 IV	Header Info	MI data	None	Success/ Fail	X		None
Transcode Audio	None	None	Header Info	Plaintext voice data	Plaintext voice data	Success/ Fail	X		None
Generate Tone	AES OFB	256 AES	Header Info	Tone parameters	MI data Plaintext or Ciphertext voice data	Success/ Fail	X		Encrypt with TEK
IMBE FEC Decode	None	None	Header Info	Plaintext voice data	Plaintext voice data	Success/ Fail	X		None
Imbe FEC Encode	None	None	Header Info	Voice data	Voice data	Success/ Fail	X		None
Encrypt Data	AES OFB	256 AES	Header Info	Plaintext data	MI data Ciphertext data	Success/ Fail	X		Generate IV Encrypt with TEK Access DRBG
Decrypt Data	AES OFB	256 AES	Header Info	MI data Ciphertext	Plaintext data	Success/ Fail	X		Decrypt with TEK

				data					
Create Random Key	DRBG	256 AES	Header Info	Key Index	None	Success/ Fail		X	Generate TEK and KEK Access DRBG
Store Key	AES Key Wrap	256 AES	Header Info	Key Index for key to store Key Index for KEK Key data	None	Success/ Fail		X	Unwrap Key with KEK Calculate HMAC with KMACK Unwrap with KSK
Delete Key	None	256 AES	Header Info	Key Index	None	Success/ Fail		X	Delete Key
Encrypt Key	AES Key Wrap	256 AES 192 AES 128 AES	Header Info	Key Index for key to encrypt Key Index for KEK	Ciphertext key data	Success/ Fail	X		Wrap Key with KEK
Calculate MAC	AES CBC	256 AES	Header Info	Key Index Plaintext data	MAC data	Success/ Fail	X		Encrypt with KMACK
Clear Key Database (Zeroize)	None	All	Header Info	None	None	Success/ Fail		X	Delete and generate KSK and KMACK Generate DRBG Seed Seed DRBG
Set Special Key	None	256 AES 192 AES 128 AES	Header Info	Key Index for existing key Key Index for copied key	None	Success/ Fail		X	Calculate HMAC with KMACK Wrap with KSK
Generate Subscriber Authentication Challenge	AES ECB DRBG	128 AES	Header Info	Key Index for existing key	DRBG output data Ciphertext data	Success/ Fail	X		Encrypt with P25SAK Access DRBG
Generate Subscriber Authentication Response	AES ECB DRBG	128 AES	Header Info	Key Index for existing key Plaintext	DRBG output data Ciphertext	Success/ Fail	X		Encrypt with P25SAK Access DRBG

				data	data				
Generate System Authentication Response	AES ECB	128 AES	Header Info	Key Index for existing key Plaintext data	Ciphertext data	Success/ Fail	X		Encrypt with P25SAK
Generate Subscriber Authentication Parameters	AES ECB DRBG	128 AES	Header Info	Key Index for existing key	DRBG output data P25 Session Key	Success/ Fail	X		Encrypt with P25SAK Access DRBG
Generate Subscriber Challenge with Parameters	AES ECB DRBG	128 AES	Header Info	P25 Session Key	DRBG output data Ciphertext data	Success/ Fail	X		Encrypt with P25SAK Access DRBG Decrypt with P25 Session Key
Generate System Response with Parameters	AES ECB	128 AES	Header Info	P25 Session Key	Ciphertext data	Success/ Fail	X		Encrypt with P25SAK Decrypt with P25 Session Key
Test Load Key	None	256 AES 192 AES 128 AES	Header Info	Key Index IV	None	Success/ Fail		X	Associate Stored Key with Stream ID
Test Encrypt	AES ECB AES CBC AES OFB	256 AES 192 AES 128 AES	Header Info	Plaintext data	Ciphertext data	Success/ Fail		X	Encrypt with TEK
Test Decrypt	AES ECB AES CBC AES OFB	256 AES 192 AES 128 AES	Header Info	Ciphertext data	Plaintext data	Success/ Fail		X	Decrypt with TEK
Test DRBG	DRBG	None	Header Info	Modulus data Seed data	DRBG output data	Success/ Fail		X	Generate DRBG Seed Seed DRBG
Test Hash	SHA-1 SHA-256 SHA-512	None	Header Info	Data to be hashed	Hash output data	Success/ Fail		X	None
Test HMAC	HMAC SHA-256	256 HMAC	Header Info	HMAC key Data to perform	MAC data	Success/ Fail		X	None

				HMAC on					
Test ECDSA	ECDSA SHA-512	Point Q on P-256 curve	Header Info	Signature parameters Data to be signed Expected signature	Signature valid/non-valid	Success/ Fail		X	None
Test RNG	None	None	Header Info	None	NDRNG debiased output data	Success/ Fail		X	None
Test Bias RNG	None	None	Header Info	None	NDRNG raw output samples	Success/ Fail		X	None
Test Key Wrap	AES Key Wrap	256-bit AES 192-bit AES 128-bit AES	Header Info	Key Plain text	Wrapped text	Success/ Fail		X	None
Test Key Unwrap	AES Key Unwrap	256-bit AES 192-bit AES 128-bit AES	Header Info	Key Wrapped text	Plain text	Success/ Fail		X	None
Continue FIPS	DRBG HMAC SHA-512	256 HMAC	Command	None	None	Success/ Fail FIPS LED		X	Generate DRNG Seed Generate KSK and KMACK Activate FIPS Approved Key Storage Calculate HMAC with KMACK Seed DRNG
Continue NonFIPS	DRBG HMAC SHA-512	256 HMAC	Command	None	None	Success/ Fail FIPS LED		X	Generate DRNG Seed Generate KSK and KMACK Activate FIPS Non-Approved Key Storage Calculate HMAC with KMACK Seed DRNG

Stop	DRBG	none	Command	None	None	Success/ Fail		X	Deactivate DRNG Deactivate Key Storage
Code Update	ECDSA SHA-256	Point Q on P-256 curve	Command	Firmware data	None	Success/ Fail		X	Verify Signature with public ECDSA key
Self-test	All	All	Not an explicit service Inherent to device Self-initiated on reset	None	None	Success/ Fail	X	X	None
Show Status	None	None	Not an explicit service Inherent to device	None	None	Success/ Fail	X	X	None

3.2.1 Load Key

The Load Key service commands the module to load an existing key from its key storage for subsequent use by other service commands. The desired key is specified by an index value that allows the module to retrieve the correct key from its key store and associate it with a stream ID for subsequent encryption or decryption operations. The module responds to this command by loading the desired key and issuing an acknowledgement, or by detecting and reporting an error.

3.2.2 Validate Key

The Validate Key service commands the module to verify the existence of a key in its key storage and report the results. The desired key is specified by an index value that allows the module to identify the correct key and examine its properties. The module responds to this command by accessing the specified key and reporting success or failure at verifying the existence of the key.

3.2.3 Get Random Data

The Get Random Data Service commands the module to generate a specified number of cryptographic-quality pseudo-random bytes and return them to the host. The module responds to this command by generating the specified number of pseudo-random bytes and returning them, or by reporting an error if it is unable to generate the pseudo-random bytes.

3.2.4 Enter Voice Tx

The Enter Voice Tx Service commands the module to prepare for voice transmission. This service must be called prior to using any of the voice transmission related services: Encrypt LC, Encrypt Voice Frame, Encrypt LSD, Transcode Audio, and Generate Tone. If Encrypt LC, Encrypt Voice Frame, or Encrypt LSD are to be used subsequently, the Load Key service must have been called prior to using this service. Use of Transcode Audio does not require a key therefore the Load Key service does not need to have been called prior. If subsequent Generate Tone service calls require encryption, the Load Key service must have been called prior to using this service. If specified in the service call, the module responds to this command by generating a 64-bit pseudo-random message indicator as specified in TIA-102.AAAD that is used as an initialization vector in the subsequent OFB encryption that follows. The module will return the generated MI if specified to the host, or report an error. This service call also specifies if the voice channel is to be in FDMA or TDMA mode.

3.2.5 Encrypt LC

The Encrypt LC service commands the module to encrypt 9 bytes of link control data. The host supplies 9 bytes of plaintext link control data that the module encrypts and returns as 9 bytes of ciphertext data. This data is encrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and Enter Voice Tx service must have been called prior to using this service. The module will return the ciphertext to the host, or report an error.

3.2.6 Encrypt Voice Frame

The Encrypt Voice Frame service commands the module to encrypt 11 bytes of IMBE digital voice data. The host supplies 20ms of audio data in one of three formats: 11 bytes of plaintext IMBE data, 160 8kHz 16-bit linear PCM audio samples, or 160 8kHz 8-bit mu-law compressed PCM audio samples. For formats other than IMBE data, the module will first encode the samples into IMBE data. The module will then encrypt the 11 bytes of plaintext IMBE data and return 11 bytes of ciphertext IMBE voice data. This data is encrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and Enter Voice Tx service must have been called prior to using this service. The module will return the ciphertext to the host, or report an error. Periodically, as specified in TIA-102.AAAD, the module will also return the next 64-bit MI to the host, along with the 11 bytes of ciphertext.

3.2.7 Encrypt LSD

The Encrypt LSD service commands the module to encrypt 2 bytes of low speed data. The host supplies 2 bytes of plaintext low speed data that the module encrypts and returns as 2 bytes of ciphertext data. This data is encrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and Enter Voice Tx service must have been called prior to using this service. The module will return the ciphertext to the host, or report an error.

3.2.8 Enter Voice Rx

The Enter Voice Rx service commands the module to prepare for voice reception. This service must be called prior to using any of the voice reception related services: Decrypt LC, Decrypt Voice Frame, Decrypt LSD, Set MI, Transcode Audio, IMBE FEC Decode, and IMBE FEC Encode. Except for the Transcode Audio service, the Load Key service must have been called prior to using this service. Use of Transcode Audio does not require a key therefore the Load Key service does not need to have been called prior. The host must supply the module with the 64-bit MI that has been received, except in the case where only Transcode Audio is called subsequently. The MI will be used as an initialization vector in the subsequent OFB decryption that follows. The module will acknowledge receipt of the MI, or report an error. This service call also specifies if the voice channel is to be in FDMA or TDMA mode.

3.2.9 Decrypt LC

The Decrypt LC service commands the module to decrypt 9 bytes of link control data. The host supplies 9 bytes of ciphertext link control data that the module decrypts and returns as 9 bytes of plaintext data. This data is decrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and

Enter Voice Rx service must have been called prior to using this service. The module will return the plaintext to the host, or report an error.

3.2.10 Decrypt Voice Frame

The Decrypt Voice Frame service commands the module to decrypt 11 bytes of IMBE digital voice data. The host supplies 11 bytes of ciphertext IMBE digital voice data that the module decrypts. This data is decrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and Enter Voice Rx service must have been called prior to using this service. The module will return plaintext data to the host in the specified format, or report an error. The returned plaintext data will be 20ms of audio data in one of three formats: 11 bytes of plaintext IMBE data, 160 8kHz 16-bit linear PCM audio samples, or 160 8kHz 8-bit mu-law compressed PCM audio samples. Periodically, as specified in TIA-102.AAAD, the host may supply a received 64-bit MI to the module, along with the 11 bytes of ciphertext.

3.2.11 Decrypt LSD

The Decrypt LSD service commands the module to decrypt 2 bytes of low speed data. The host supplies 2 bytes of ciphertext low speed data that the module decrypts and returns as 2 bytes of plaintext data. This data is decrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service and Enter Voice Rx service must have been called prior to using this service. The module will return the plaintext to the host, or report an error.

3.2.12 Set MI

The Set MI service commands the module to set the 64-bit MI to the value supplied by the host. This service provides an alternate way to load the MI, which is customarily loaded using the Enter Voice Rx service or Decrypt Voice Frame service. The module will accept the supplied MI and return and acknowledgement to the host, or report an error.

3.2.13 Transcode Audio

The Transcode Audio service commands the module to convert audio data from one format to another format. This service does not provide encryption or decryption functionality. The host supplies 20ms of audio data in one format and specifies the desired format of the returned audio data. The allowed formats are: 11 bytes of plaintext IMBE data, 160 8kHz 16-bit linear PCM audio samples, or 160 8kHz 8-bit mu-law compressed PCM audio samples. The module will return 20ms of audio data in the specified format, or report an error. When converting from IMBE data to IMBE data it is implied that the IMBE data should be transcoded between FDMA and TDMA given the current mode of the stream.

3.2.14 Generate Tone

The Generate Tone service commands the module to generate a 20ms frame of IMBE digital voice data containing a specified tone which is optionally encrypted. If specified, the module will encrypt the 11 bytes of generated IMBE data using AES in OFB mode as specified in TIA-102.AAAD. If encryption is to be performed, the Load Key service must have been called prior to using this service. The Enter Voice Tx service must always be called prior to using this service. The module will return the plaintext or ciphertext to the host, or report an error. Periodically, as specified in TIA-102.AAAD, the module will also return the next 64-bit MI to the host, along with the 11 bytes of ciphertext.

3.2.15 IMBE FEC Decode

The IMBE FEC Decode service commands the module to perform FEC decode on a raw IMBE voice frame. This service does not provide encryption or decryption functionality. The host supplies the FEC decode options along with a 144 byte IMBE voice frame (each soft-byte represents one bit). The module will return the 11 bytes of plaintext IMBE data with the 2 byte FEC info, or report an error.

3.2.16 IMBE FEC Encode

The IMBE FEC Encode service commands the JEM2 to perform FEC encode on a raw IMBE voice frame. This service does not provide encryption or decryption functionality. The host supplies the 11 byte IMBE voice frame. The module will return the 144 bytes (each soft-byte represents one bit) of IMBE data, or report an error.

3.2.17 Encrypt Data

The Encrypt Data service commands the module to encrypt a variable amount of data. The host supplies a variable amount of plaintext data (along with the length of the data) that the module encrypts and returns as ciphertext data. This data is encrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service must have been called prior to using this service. The host can optionally request the module to generate a 64-bit MI to accompany the ciphertext. The module will return the ciphertext and the optional MI to the host, or report an error.

3.2.18 Decrypt Data

The Decrypt Data service commands the module to decrypt a variable amount of data. The host supplies a variable amount of ciphertext data (along with the length of the data) that the module decrypts and returns as plaintext data. This data is decrypted using AES in OFB mode as specified in TIA-102.AAAD. The Load Key service must have been called prior to using this service. The host can optionally supply the module with a 64-bit MI to accompany the ciphertext. The module will return the plaintext to the host, or report an error.

3.2.19 Create Random Key

The Create Random Key service commands the module to generate a 256 bit AES pseudo-random key using the module’s DRBG and place it into its key store. The host provides an index value that instructs the module where to store the key. The module responds to this command by generating a 256 bit AES pseudo-random key, placing it in the key store and issuing an acknowledgement, or by reporting an error. The seed generation for the DRBG is discussed in the critical security parameters section.

3.2.20 Store Key

The Store Key service commands the module to accept a (plaintext or ciphertext) key from the host and place it into the key store. The host supplies a key and an index that tells the module where to store the key. If the key is encrypted with the AES Key Wrap algorithm, the host also supplies the index of a key encryption key that the module will use to decrypt the encrypted key before placing it into the key store. The module responds to this command by (decrypting an encrypted key), placing it into the key store and issuing an acknowledgement, or by reporting an error.

3.2.21 Delete Key

The Delete Key service commands the module to delete the key specified by the index provided by the host. The module responds to this command by deleting the specified key from the key store and issuing an acknowledgement, or by reporting an error.

3.2.22 Encrypt Key

The Encrypt Key service commands the module to retrieve a key from the key store, encrypt the key and supply the encrypted key to the host. The host supplies an index that tells the module where to retrieve the key. The host also supplies the index of a key encryption key that the module will use to encrypt the retrieved key before sending it to the host. The module responds to this command by retrieving the specified key from the key store, encrypting the key with the AES Key Wrap algorithm using the specified KEK and supplying the encrypted key to the host, or by reporting an error.

3.2.23 Calculate MAC

The Calculate MAC service commands the module to accept a variable length message and a key index and use the specified key to compute a 64-bit MAC of the message. The module responds to this command by using the specified key and the AES cipher in CBC mode to compute the MAC as specified in TIA-120.AACA-1. Once calculated, the module returns the 64-bit MAC to the host, or reports an error. Note that this algorithm is only allowed to be used within OTAR.

3.2.24 Clear Key Database (Zeroize)

The Clear Key Database service commands the module to zeroize all keys and CSPs within the cryptographic boundary of the module.

3.2.25 Set Special Key

The Set Special Key service commands the module to create a copy of an existing key in its key store. The host supplies an index for the existing key to copy and an index of where to store the copy. The module responds to this command by copying the existing key to the new index and returning an acknowledgement to the host, or by reporting an error.

3.2.26 Generate Subscriber Authentication Challenge

The Generate Subscriber Authentication Challenge service commands the module to create a P25 subscriber challenge and its expected response. The host supplies an index for the existing key to be used for this operation. The module responds to this command with the subscriber challenge, the expected response, and the random seed data used to generate the subscriber challenge, or report an error.

3.2.27 Generate Subscriber Authentication Response

The Generate Subscriber Authentication Response service commands the module to find the P25 subscriber response as well as generate a P25 system challenge. The host supplies the subscriber challenge and random seed along with an index for the existing key to be used. The module responds to this command with the subscriber response, a system challenge, and the expected system response, or report an error.

3.2.28 Generate System Authentication Response

The Generate System Authentication Response service commands the module to find the P25 system response. The host supplies the system challenge and random seed along with an index for the existing key to be used. The module responds to this command with the system response, or report an error.

3.2.29 Generate Subscriber Authentication Parameters

The Generate Subscriber Authentication Parameters service commands the module to create the parameters required to perform P25 System Authentication. The host supplies an index for the existing key to be used for this operation. The module responds to this command with the 10 byte random seed, the 16 byte subscriber parameter and the 16 byte system parameter, or report an error.

3.2.30 Generate Subscriber Challenge with Parameters

The Generate Subscriber Challenge with Parameters service commands the module to create a P25 subscriber challenge and its expected response. The host supplies the subscriber parameters to be used for this operation. The module responds to this command with the subscriber challenge and the expected response, or report an error.

3.2.31 Generate System Response with Parameters

The Generate System Response with Parameters service commands the module to find the P25 system response. The host supplies the system challenge and the system parameters to be used. The module responds to this command with the system response, or report an error.

3.2.32 Test Load Key

The Test Load Key service commands the module to load an existing key from its key storage for use in algorithm tests. The desired key is specified by an index value that allows the module to retrieve the correct key from its key store and associate it with a stream ID for subsequent encryption or decryption operations. The host also specifies the mode of operation and an initialization vector. The module responds to this command by loading the desired key and issuing an acknowledgement, or by reporting an error.

3.2.33 Test Encrypt

The Test Encrypt service commands the module to encrypt 16 bytes of test data. The host supplies 16 bytes plaintext data that the module encrypts and returns as ciphertext data. The Test Load Key service must have been called prior to using this service. The module will return the ciphertext to the host, or report an error.

3.2.34 Test Decrypt

The Test Decrypt service commands the module to decrypt 16 bytes of test data. The host supplies 16 bytes of ciphertext data that the module decrypts and returns as plaintext data. The Test Load Key service must have been called prior to using this service. The module will return the plaintext to the host, or report an error.

3.2.35 Test DRBG

The Test DRBG Service commands the module to generate a cryptographic-quality pseudo-random number and return it to the host. The host supplies parameters for seeding and iterating the DRBG. The module responds to this command by generating a pseudo-random number and returning it, or by reporting an error if it is unable to generate the pseudo-random number.

3.2.36 Test Hash

The Test Hash service commands the module to generate a hash and return it to the host. The host specifies the hash algorithm to use (SHA-1, SHA-256, or SHA-512) and supplies the data. The module will return the hash to the host, or report an error.

3.2.37 Test HMAC

The Test HMAC service commands the module to generate an HMAC (using SHA-256) and return it to the host. The host supplies the key and data to perform the HMAC algorithm on. The module will return the HMAC to the host, or report an error.

3.2.38 Test ECDSA

The Test ECDSA service commands the module to verify a digital signature. The host supplies the parameters for the ECDSA algorithm, the text, and signature. The module will verify the signature. The module will return whether the signature is verified, or report an error.

3.2.39 Test RNG

The Test RNG service commands the module to retrieve random data from the module's hardware NDRNG and return them to the host. The module responds to this command by retrieving raw samples from the hardware NDRNG and processing those samples into random bits and returning them, or by reporting an error if it is unable to generate the random data.

3.2.40 Test Bias RNG

The Test Bias RNG Service commands the module to retrieve raw (unprocessed) samples data from the module's hardware NDRNG and return them to the host. The module responds to this command by retrieving raw samples from the hardware NDRNG and returning them, or by reporting an error if it is unable to generate the random data.

3.2.41 Test Key Wrap

The Test Key Wrap service commands the module to wrap data with a given key. The host supplies the parameters for the key and the unwrapped plain text. The module will return the status of success and the wrapped cipher text, or report an error.

3.2.42 Test Key Unwrap

The Test Key Unwrap service commands the module to unwrap data with a given key. The host supplies the parameters for the key and the wrapped cipher text. The module will return the status of success and the unwrapped plain text, or report an error.

3.2.43 Continue FIPS

The Continue FIPS service commands the module to enter its FIPS approved mode of operation. The module responds to this service by going into its FIPS approved mode of operation. The module will signal the host that it has successfully entered its FIPS approved mode of operation.

3.2.44 Continue NonFIPS

The Continue NonFIPS service commands the module to enter its FIPS non-approved mode of operation. The module responds to this service by going into its FIPS non-approved mode of operation. The module will signal the host that it has successfully entered its FIPS non-approved mode of operation.

3.2.45 Stop

The Stop service commands the module to go into a stopped state. The module responds to this service by going into a stopped state where it will only process the Continue FIPS, Continue NonFIPS, and Code Update services. The module will signal the host that it successfully entered its stopped state.

3.2.46 Code Update

The Code Update service allows the crypto officer to provide the module with a digitally signed firmware update that the module will verify and authenticate. If the signature verifies, the JEM is in FIPS mode of operation, and the firmware update is authentic, then the module will update its firmware as requested and report success. If the signature fails to verify, the JEM2 is not in FIPS mode, or if the operation does not successfully complete, the module will report an error. Note only EF Johnson can generate valid signatures.

3.2.47 Self-Test

The Self-Test service commands the module to perform the power-up tests to verify proper secure operation. All module cryptographic and security functions are tested using known answer tests. When the self-tests are complete the module will report success or failure.

Note: The Self-Test Service is initiated by cycling power or resetting the module.

3.2.48 Show Status

The Show Status Service commands the module to display its current status and mode of operation. Status is returned at completion of each service. The status LED also indicates the status of the device.

Note: The Show Status Service is inherent in the operation of the module and is not an explicit service.

3.3 Services and CSP Relationships

Table 10 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as:

- G = Generate: The module generates the CSP.
- R = Read: The module reads the CSP. The read access is typically performed before the module uses the CSP.
- E = Execute: The module executes using the CSP.
- W = Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module, when the module generates a CSP, or when the module overwrites an existing CSP.
- Z = Zeroize: The module zeroizes the CSP.

Table 10 – CSP Access Rights within Services

Service	CSPs							
	Key Storage Key (KSK)	Key MAC Key (KMACK)	Traffic Encryption Key (TEK)	Key Encryption Key (KEK)	DRBG Working State	P25 System Authentication Key (P25SAK)	P25SSAK	P25SMSAK
Load Key	R/E	-	R	-	-	-	-	-
Validate Key	-	-	R	-	-	-	-	-
Get Random Data	-	-	-	-	R/W/E	-	-	-
Enter Voice Tx	-	-	-	-	R/W/E	-	-	-
Encrypt LC	-	-	E	-	-	-	-	-
Encrypt Voice Frame	-	-	E	-	-	-	-	-
Encrypt LSD	-	-	E	-	-	-	-	-

Service	CSPs							
	Key Storage Key (KSK)	Key MAC Key (KMACK)	Traffic Encryption Key (TEK)	Key Encryption Key (KEK)	DRBG Working State	P25 System Authentication Key (P25SAK)	P25SSAK	P25SMSAK
Enter Voice Rx	-	-	-	-	-	-	-	-
Decrypt LC	-	-	E	-	-	-	-	-
Decrypt Voice Frame	-	-	E	-	-	-	-	-
Decrypt LSD	-	-	E	-	-	-	-	-
Set MI	-	-	-	-	-	-	-	-
Transcode Audio	-	-	-	-	-	-	-	-
Generate Tone	-	-	E	-	-	-	-	-
IMBE FEC Decode	-	-	-	-	-	-	-	-
IMBE FEC Encode	-	-	-	-	-	-	-	-
Encrypt Data	-	-	E	-	R/W/E	-	-	-
Decrypt Data	-	-	E	-	-	-	-	-
Create Random Key	R/E	R/E	G	G	R/W/E	-	-	-
Store Key	R/E	R/E	W	-	-	-	-	-
Delete Key	-	-	Z	-	-	-	-	-
Encrypt Key	R/E	-	R/W	R/E	-	-	-	-
Calculate MAC	R/E	-	R	-	-	-	-	-
Clear Key Database (Zeroize)	Z/G/W	Z/G/R	Z	Z	Z/G	Z	Z	Z
Set Special Key	R/E	R/E	R/G	-	-	-	-	-
Generate Subscriber Authentication Challenge	R/E	-	E	-	R/W/E	R/W/E	R/W/E	-
Generate Subscriber Authentication Response	R/E	-	E	-	R/W/E	R/W/E	R/W/E	-
Generate System Authentication Response	R/E	-	E	-	-	R/W/E	-	R/W/E
Generate Subscriber Authentication Parameters	R/E	-	-	-	R/W/E	R/W/E	G	G
Generate Subscriber Challenge with Parameters	-	-	-	-	R/W/E	-	R/E	-
Generate System Response with Parameters	-	-	-	-	-	-	-	R/E
Test Load Key	R/E	-	R	-	-	-	-	-
Test Encrypt	-	-	E	-	-	-	-	-
Test Decrypt	-	-	E	-	-	-	-	-
Test DRBG	-	-	-	-	Z/G/R/W/E	-	-	-
Test Hash	-	-	-	-	-	-	-	-
Test HMAC	-	-	R	-	-	-	-	-
Test ECDSA	-	-	-	-	-	-	-	-
Test RNG	-	-	-	-	-	-	-	-
Test Bias RNG	-	-	-	-	-	-	-	-

Service	CSPs							
	Key Storage Key (KSK)	Key MAC Key (KMACK)	Traffic Encryption Key (TEK)	Key Encryption Key (KEK)	DRBG Working State	P25 System Authentication Key (P25SAK)	P25SSAK	P25SMSAK
Test Key Wrap	-	-	-	-	-	-	-	-
Test Key Unwrap	-	-	-	-	-	-	-	-
Continue FIPS	G/R/W	G/R/E	W	-	G	-	-	-
Continue NonFIPS	G/R/W	G/R/E	W	-	G	-	-	-
Stop	-	-	W	-	Z	-	-	-
Code Update	-	-	-	-	-	-	-	-
Self-Test	-	-	-	-	G/R/W/E/Z	-	-	-
Show Status	-	-	-	-	-	-	-	-

4 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 self-tests described in Table 11 below. All KATs must be completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the Fatal Error state.

Table 11 – Power Up Self-tests

Test Target	Description
Firmware Integrity	SHA-512 performed over all code in FLASH.
AES	KATs: Encryption, Decryption Modes: ECB Key sizes: 128 bits, 192 bits, 256 bits
AES-KW	KATs: Wrap, Unwrap Mode: KW Key sizes: 128 bits, 192 bits, 256 bits
DRBG	KATs: HASH DRBG Security Strengths: 512 bits • Health Testing accomplished by KAT
ECDSA	PCT: Signature Verification Curves/Key sizes: P-256
HMAC	KATs: Generation, Verification SHA sizes: SHA-256

Test Target	Description
SHA	KATs: SHA-1, SHA-256, SHA-512

Table 12 – Conditional Self-tests

Test Target	Description
NDRNG	NDRNG Continuous Test performed when a random value is requested from the NDRNG.
DRBG	DRBG Continuous Test performed when a random value is requested from the DRBG.
Firmware Load	ECDSA P-256 signature verification performed when firmware is loaded.

Table 13 – Critical Functions Tests

Test Target	Description
Memory	The RAM is tested by checking that all zeros and all ones can be written into each word.

5 Physical Security Policy

The Module is a multi-chip embedded cryptographic module that uses commercial-grade production components. Since the Module is designed to meet the requirements of FIPS 140-2 Security Level 1, it does not employ environmental failure protection (EFP) features nor will it undergo environmental failure testing (EFT). The operator is not required to perform any physical security inspections.

6 Operational Environment

The Module is designated as a limited operational environment under the FIPS 140-2 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 140-2 CMVP. Any other firmware loaded into this module is out of the scope of this validation and require a separate FIPS 140-2 validation.

7 Mitigation of Other Attacks Policy

The module is not designed to mitigate against other attacks not specifically mentioned in this document, including but not limited to power analysis, timing analysis, fault indication or TEMPEST.

8 Security Rules and Guidance

The Module design corresponds to the Module security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

8.1 FIPS 140-2 Related Security Rules

1. The module shall provide two distinct operator roles: User and Cryptographic Officer. The role is selected implicitly by the service that is invoked.
2. When the module has not been placed in a valid role, the operator shall not have access to any cryptographic services.
3. The operator shall be capable of commanding the module to perform the power up self-tests by cycling power or resetting the module.
4. Power up self-tests do not require any operator action.
5. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
6. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
7. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
8. The module does not support concurrent operators.
9. The module does not support a maintenance interface or role.
10. The module does not support manual key entry.
11. The module does not have any external input/output devices used for entry/output of data.
12. The module does not output plaintext CSPs.
13. The module does not output intermediate key values.

8.2 EFJohnson Technologies Imposed Security Rules

1. The module is incorporated into a host computer that is part of a Project 25 infrastructure. An authorized user of the host has the ability to select either a FIPS 140-2 approved mode of operation or a non-approved mode of operation for the module. CSPs are segregated for approved and non-approved modes of operation, and only the applicable set of CSPs are in use at any given time.
2. The module FIPS 140-2 approved mode of operation provides AES cryptographic services for current Project 25 infrastructure systems. The module non-approved mode of operation provides DES cryptographic services for legacy Project 25 infrastructure systems.
3. When operating in the FIPS 140-2 approved mode, the module uses the AES block cipher as specified in FIPS-197 for encryption and decryption of voice and data, and AES CBC MAC.
4. When operating in the FIPS 140-2 approved mode, the module uses the AES Key Wrap Algorithm as specified in SP 800-38F for encryption and decryption of cryptographic keys.
5. When operating in the FIPS 140-2 approved mode, the module uses the DRBG to generate cryptographic-quality pseudo-random numbers.
6. When operating in the FIPS 140-2 approved mode, the module uses the SHA-256 hash function as specified in FIPS 180-4 and the ECDSA signature algorithm to verify digital signatures of firmware updates.
7. When operating in the non-approved mode, the module uses the DES block cipher for encryption and decryption of voice and data.

8. The module provides for storage of keys (key storage key, traffic encryption keys, key encryption keys, and P25 system authentication keys) in plaintext form within the cryptographic boundary of the module.
9. The module has an LED output indicator that informs the user of the status of the crypto module and its operating mode (FIPS 140-2 approved or non-approved).

9 References and Definitions

The following standards are referred to in this Security Policy.

Table 14 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, May 25, 2001
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011
[SP800-57]	*Recommendation for Key Management*, Part 1 July 2012, Part 2 August 2005, Part 3 December 2009
[SP800-90A]	*Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, January 2012
[SP800-107]	*Recommendation for Applications Using Approved Hash Algorithms,* August 2012
[SP800-133]	*Recommendation for Cryptographic Key Generation, Revision 2,* June 2020
[SP800-38F]	*Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping,* December 2012
[FIPS186-3]	*Digital Signature Standard*, June 2009
[FIPS198-1]	*The Keyed-Hash Message Authentication Code*, July 2008
TIA-AACA	*APCO Project 25, Over-The-Air-Rekeying (OTAR) Protocol, New Technology Standards Project, Digital Radio Technical Standards*, TSB102.AACA
TIA-AACE-A	*Digital Land Mobile Radio Link Layer Authentication,* April 2011.
TIA P25 Standard	*Digital Radio Over-the-Air-Rekeying (OTAR) Protocol Addendum 1 – Key Management Security Requirements for Type 3 Block Encryption Algorithms*.

Table 15 – Acronyms and Definitions

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
DCB	Device Control Block
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
DSP	Digital Signal Processor
DSS	Digital Signature Standard

Acronym	Definition
DTR	Derived Test Requirements
ECB	Electronic Codebook
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	Error Detection Code
EFP	Environmental Failure Protection
EFT	Environmental Failure Testing
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESYNC	Encryption Synchronization
FCC	Federal Communications Commission
FDMA	Frequency-Division Multiple Access (as defined by P25 Standard)
FIPS	Federal Information Processing Standards
HMAC	Keyed-Hashing for Message Authentication
IC	Integrated Circuit
IMBE	Improved Multi-Band Exciter (P25 Standard Vocoder)
IV	Initialization Vector
JEM2	Johnson Encryption Machine 2
KAT	Known Answer Test
KEK	Key Encryption Key
KMACK	Key Message Authentication Code Key
KSK	Key Storage Key
LC	Link Control
LED	Light Emitting Diode
LSD	Low Speed Data
MAC	Message Authentication Code
MI	Message Indicator
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
OFB	Output-Feedback
OTAR	Over-The-Air-Rekeying
P25	Project 25

Acronym	Definition
P25SAK	P25 System Authentication Key
P25SSAK	P25 System Session Authentication Key
P25SMSAK	P25 System Mutual Session Authentication Key
PCB	Printed Circuit Board
PCIe	Peripheral Component Interconnect (express)
PIN	Personal Identification Number
ROM	Read Only Memory
RAM	Random Access Memory
RFC	Request For Comments
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
TIA	Telecommunication Industry Association
TEK	Transmission Encryption Key