Security Policy: IPCryptR2

Cryptographic module used in Motorola's IPCryptR2 for Astro, Dimetra and Broadband systems

Version: R01.00.23

Date: February 23, 2017

Table of Contents

1. Introduction

1.1. Scope

This Security Policy specifies the security rules under which the IPCryptR2 must operate. In addition to the security requirements derived from FIPS 140-2 are those imposed by Motorola Solutions, Inc. (Motorola). These rules, in total, define the interrelationship between the:

- Module Operators,
- Module Services, and
- Critical Security Parameters (CSPs).

1.2. Definitions

ALGID	Algorithm Identifier
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CKR	Common Key Reference
CSP	Critical Security Parameter
DES	Data Encryption Standard
ECB	Electronic Code Book
ECDH	Elliptic Curve Diffie-Hellman
IKE	Internet Key Exchange
IPSec	Internet Protocol security
ISAKMP	Internet Security Association and Key Management Protocol
IV	Initialization Vector
KLK	Key Loss Key
KPK	Key Protection Key
KVL	Key Variable Loader
LED	Light-emitting diode
NDRNG	Non-deterministic Random Number Generator
PEK	Password Encryption Key
RAM	Random Access Memory
RNG	Random Number Generator

1.3. Overview

The IPCryptR2 provides secure key management and data encryption in Astro, Dimetra and Broadband Systems.

1.4. IPCryptR2 Implementation

The IPCryptR2 is implemented as a multi-chip standalone cryptographic module as defined by FIPS 140-2.

1.5. IPCryptR2 Hardware / Firmware Version Numbers

The IPCryptR2 has the following FIPS validated hardware and firmware version numbers:

Table 1: FIPS Validated Version Numbers

FIPS Validated Cryptographic Module Hardware Kit Numbers	FIPS Validated Cryptographic Module Firmware Version Numbers
BLN1306A	R06.03.05

1.6. IPCryptR2 Cryptographic Boundary

The IPCryptR2 cryptographic boundary is drawn around the entire product which includes the housing, various IC's, FLASH, RAM, and Printed Circuit Board as shown below.

Front (Port Covers Closed)

Front (Port Covers Open)

Top

Rear (Port Covers Closed)

Rear (Port Covers Open)

Figure 1: IPCryptR2 Product Diagram

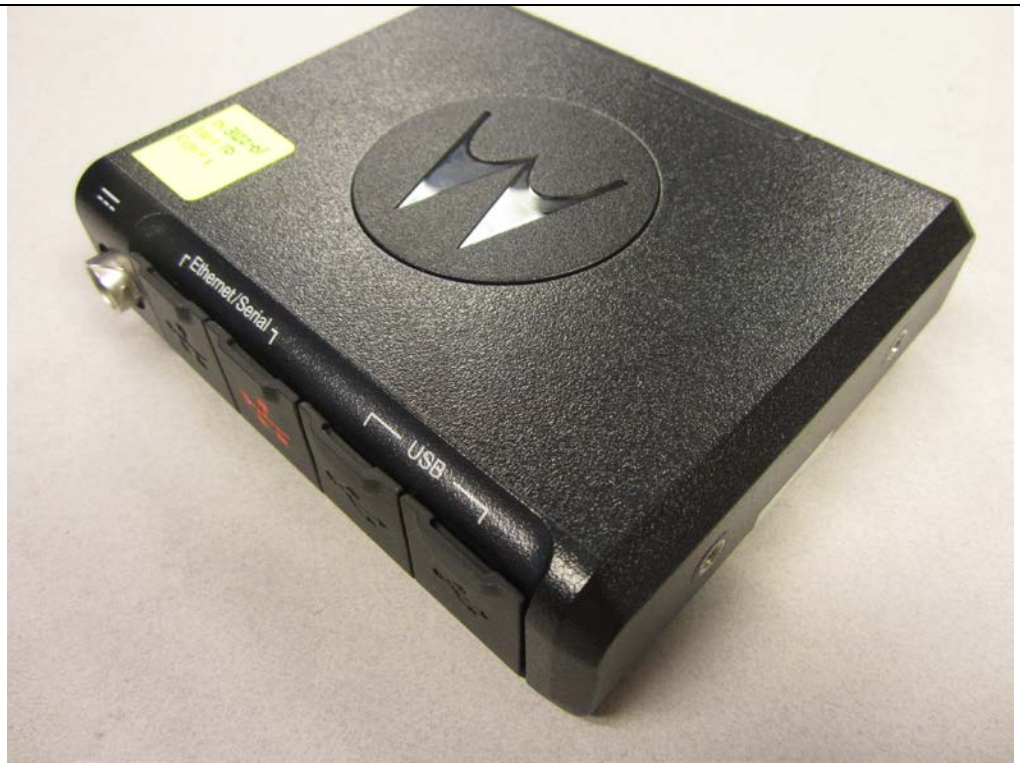

Figure 2: IPCryptR2 Module (Top/Front/Right)

Figure 3: IPCrytpR2 Module (Bottom/Left/Rear)

1.7. Ports and Interfaces

The IPCryptR2 provides the following physical ports and logical interfaces:

Table 2: Ports and Interfaces

Physical Port	Qty	Logical Interface Definition	Description
Power	1	Power Input	This interface powers all circuitry. This interface does not support input / output of CSPs.
Key Variable Loader (KVL) Interface	1	Data Input Data Output Control Input Status Output	Provides an interface to the Key Variable Loader. KEKs and TEKs are entered in encrypted form over the KVL interface. The hash of the boot block is output over the KVL interface if the Firmware Integrity Test is successful on power up. This interface does not support output of CSPs.
Key Variable Loader (KVL) Auxiliary Interface	1	N/A	This interface is not used by the module. This interface is disabled.
RS-232 Interface	1	Control Input Status Output Data Output	Provides an interface for factory programming and execution of RS-232 shell commands. This interface does not support output of CSPs.
Mini-Universal Serial Bus (mini-USB) Interface	1	Control Input Status Output Data Output	Provides an interface for execution of shell commands. This interface does not support output of CSPs.
Ethernet Interface	2	Data Input Data Output Control Input Status Output	This interface routes packets between subnets. The IP stack of this interface will use the subnet information to determine how to route packets between physical network interfaces. The red Ethernet interface is for input and output of plaintext data. The black Ethernet interface is for encrypted data (ESP) and key establishment protocol (IKE). This interface does not support any other input / output of CSPs. This interface is used for OTAR.
Erase Switch	1	Control Input	This interface is used for zeroization of KEKs, TEKs, KPK, ECDH private/public key pair, and IPSec Session keys.
Reset	1	Control Input	This interface forces a reset of the module.

Physical Port	Qty	Logical Interface Definition	Description
Switch			
Alarm LED	1	Status Output	The Alarm LED turns red to indicate a critical error has been detected and flashing red to indicate a security condition has been detected that requires operator intervention.
Power LED	1	Status Output	The Power LED turns steady green after power is applied and flashing green to indicate a low or dead battery.
Ready LED	2	Status Output	The Ready LED turns steady green to indicate an Ethernet link has been established and is flashing green when there is activity on the link. On the black side, if DHCP is enabled and an IP Address has been acquired, the LED turns green.
TX Clear LED	1	Status Output	The TX Clear LED is not used and remains off other than during power up self-test when the LED turns green.
Status LED	1	Status Output	The Status LED is steady green when an IPSec tunnel has been established; steady red when no TEK has been loaded; and off when a TEK has been loaded but an IPSec tunnel has not been established.

2. FIPS 140-2 Security Levels

The IPCryptR2 can be configured to operate at FIPS 140-2 overall Security Level 2. The table below shows the FIPS 140-2 Level of security met for each of the eleven areas specified within the FIPS 140-2 security requirements.

Table 3: IPCryptR2 Security Levels

FIPS 140-2 Security Requirements Section	Validated Level at overall Security Level 2
Cryptographic Module Specification	2
Module Ports and Interfaces	2
Roles, Services, and Authentication	2
Finite State Model	2
Physical Security	2
Operational Environment	N/A
Cryptographic Key Management	2
EMI / EMC	2
Self-Tests	2
Design Assurance	3
Mitigation of Other Attacks	N/A

3. FIPS 140-2 Approved Operational Modes

The IPCryptR2 can be configured to operate in a FIPS 140-2 Approved mode of operation and a non-FIPS Approved mode of operation. Documented below are the configuration settings that are required for the module to be used in a FIPS 140-2 Approved mode of operation at overall Security Level 2.

1. Enable FIPS mode. When FIPS mode is enabled, the module will not allow keys to be entered in plaintext form; all keys entered into the module must be encrypted. The Configure IPCryptR2 service is used to configure this parameter in the module via the 'fips enable' shell command.
2. Only Approved and Allowed algorithms installed. The module supports the following Approved algorithms:
   - AES-256 ECB (Cert. #1424)
   - AES-256 8-bit CFB8 (Cert. #1424) – for symmetric encryption / decryption of keys and parameters stored in the internal database.
   - AES-256 OFB (Cert. #1424) – for symmetric decryption of keys entered via the KVL interface. Keys are sent to the module from the KVL interface but never uploaded back to the KVL, hence the module does not perform encryption in this mode.
   - AES-256 CBC (Cert. #1424) – for use with IKE and OTAR.
   - AES-256 GCM (Cert. #1425) – for high-speed encryption in the GCM mode.
   - CVL (Cert. #262) – IKEv2 KDF is used to set up a security association (SA) in the IPsec protocol suite, per Internet Key Exchange Protocol Version 2 (IKEv2) and Suite B Cryptographic Suites for IPsec.
   - CVL (Cert. #263) – SNMP v3 key derivation function per NIST Special Publication 800-135, section 5.4. SNMPv3 KDF is used when CRYPTR2 is being monitored remotely by an SNMP Server.
   - ECDSA-384 (Cert. #498) – used for digital signature verification during firmware integrity test and firmware load test
   - HMAC SHA-384 (Cert. #1780) – used in IKE v2 key derivation function per Internet Key Exchange Protocol Version 2 (IKEv2) and Suite B Cryptographic Suites for IPsec.
   - KTS (AES Cert. #1424 and HMAC SHA-384 Cert. #1780, key wrapping; key agreement methodology provides 256 bits of encryption strength)
   - SHA-1 (Cert. #2381) – hash algorithm for use in SNMPv3 protocol only.
   - SHA-384 (Cert. #2381) – used for password hashing for internal password storage, used for digital signature verification during firmware integrity test and firmware load test, and used as data origin authentication and integrity verification mechanisms for IKE.

The following non-Approved algorithms and protocols are allowed within the Approved mode of operation:

- AES MAC (AES Cert. #1424, vendor affirmed; P25 AES OTAR)
- EC Diffie-Hellman (key agreement; key establishment methodology provides 192 bits of encryption strength) – Asymmetric algorithms used for establishing secure private communication between two parties.
- Non-deterministic Hardware Random Number Generator, NSA approved – used for IV and key generation

The module maintains FIPS mode status and will provide this upon operator request. All functions that are available in FIPS Approved mode are also available in non-FIPS Approved mode. CSPs are not shared between FIPS Approved mode and non-FIPS Approved mode. The transition from a FIPS Approved mode to a non-FIPS Approved mode causes all CSPs to be zeroized.

4. Crypto Officer and User Guidance

4.1. Administration of the IPCryptR2 in a secure manner (CO)

The IPCryptR2 requires no special administration for secure use after it is set up for use in a FIPS Approved manner. To do this, configure the module as described in section 3 of this document.

Note that all keys can be zeroized after the Program Update service has completed. Alternatively, keys can be preserved following the Program Update service. The option to either preserve or erase the keys is selected when starting the Program Update service.

4.2. Assumptions regarding User Behavior

The IPCryptR2 has been designed in such a way that no special assumptions regarding User Behavior have been made that are relevant to the secure operation of the unit.

4.3. Approved Security Functions, Ports, and Interfaces available to Users

IPCryptR2 services available to the User role are listed in section 8.2.

4.4. User Responsibilities necessary for Secure Operation

No special responsibilities are required of the User for secure operation of the IPCryptR2.

5. Security Rules

The IPCryptR2 enforces the following security rules. These rules are separated into those imposed by FIPS 140-2 and those imposed by Motorola.

5.1. FIPS 140-2 Imposed Security Rules

1. The IPCryptR2 inhibits all data output via the data output interface whenever an error state exists and during self-tests.
2. The IPCryptR2 logically disconnects the output data path from the circuitry and processes when performing key generation or key zeroization.
3. Authentication data is not output during entry.
4. Secret cryptographic keys are entered in encrypted form over a physically separate port.
5. The IPCryptR2 enforces Role-Based authentication. Multiple concurrent operators are not supported.
6. The IPCryptR2 supports a User role and a Crypto-Officer role. The module will verify the authorization of the operator to assume each role.
7. The IPCryptR2 re-authenticates an operator when it is powered-up after being powered-off.
8. The IPCryptR2 implements all firmware using a high-level language, except the limited use of low-level languages to enhance performance.
9. The IPCryptR2 protects secret keys and private keys from unauthorized disclosure, modification and substitution.
10. The IPCryptR2 provides a means to ensure that a key entered into or stored within the module is associated with the correct entities to which the key is assigned. Each key in the IPCryptR2 is entered encrypted and stored with the following information:
    - Key Identifier – 16 bit identifier
    - Algorithm Identifier – 8 bit identifier
    - Key Type – Traffic Encryption Key or Key Encryption Key
    - Physical ID, Common Key Reference (CKR) number, and Keyset number – Identifiers indicating storage locations.

    Along with the encrypted key data, this information is stored in a key record that includes a CRC over all fields to protect against data corruption. When used or deleted the keys are referenced by CKR / Key ID / Algid, Key ID / Algid, Physical ID, or CKR / Keyset.
11. The IPCryptR2 denies access to plaintext secret and private keys contained within the module.
12. The IPCryptR2 provides the capability to zeroize all plaintext cryptographic keys and other unprotected critical security parameters within the module.
    1. To completely initialize and zeroize the module:
        i. Utilize the Program Update service, and reset the module, to zeroize all module plaintext CSPs.
13. The IPCryptR2 conforms to FCC 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B requirements.
14. The IPCryptR2 performs the following self-tests. Powering the module off then on or

resetting the module using the Reset service will initiate the power up self-tests.

- Power up and on-demand tests
  - Cryptographic algorithm test: Each algorithm (SHA-384, SHA-1, HMAC-SHA384, ECDSA-SHA384 and AES-256 in OFB, GCM, ECB, CBC, and 8-bit CFB modes) is tested by using a known key, known data, and if required a known IV. The data is then encrypted and compared with known encrypted data; the test passes if the final data matches the known data, otherwise it fails. The encrypted data is then decrypted and compared with the original plaintext; the test passes if the decrypted data matches the original plaintext, otherwise it fails. The complete list of algorithm tests follows:
    - SHA1 hashing KAT on a 24-bit data block (Cert, #2381)
    - SHA1 hashing KAT on a 448-bit data block (Cert. #2381)
    - SHA384 hashing KAT on a 24-bit block (Cert. #2381)
    - SHA384 hashing KAT on a 896-bit block (Cert. #2381)
    - HMAC SHA384 keyed hashing KAT w/25-byte key and 50-byte data (Cert. #1780)
    - AES CFB encrypt/decrypt KAT using a 128-bit key (Cert. #1424)
    - AES CFB encrypt/decrypt KAT in 8 bit mode w/a 256-bit key (Cert. #1424)
    - AES encrypt/decrypt KAT in ECB mode using a 256-bit key (Cert. #1424)
    - AES encrypt/decrypt KAT in CBC mode using a 256-bit key (Cert. #1424)
    - AES encrypt/decrypt KAT in OFB mode using a 256-bit key (Cert. #1424)
    - AES GCM encrypt/decrypt KAT using a 256-bit key (Cert. #1425)
  - Non-deterministic Hardware Random Number Generator entropy test.
  - Firmware integrity test: A digital signature is generated over the code when it is built using SHA-384 and ECDSA-384 and is stored with the code upon download into the module. When the module is powered up the digital signature is verified. If the digital signature matches, then the test passes, otherwise it fails.
  - External indicators test: Upon every power up, the module will assert and de-assert each signal connected to an external indicator, so that the User may verify that the indicators are functioning and controlled by the module.
- Conditional tests
  - Firmware load test: A digital signature is generated over the code when it is built using SHA-384 and ECDSA-384. Upon download into the module, the digital signature is verified. If the digital signature matches, then the test passes, otherwise it fails.
  - Continuous Random Number Generator test: The continuous random number generator test is performed on the RNG supported by the module. (Hardware NDRNG). An initial value is generated and stored upon power up. This value is not used for anything other than to initialize comparison data. A successive call to the RNG generates a new set of data, which is compared to the comparison data. If a match is detected, this test fails; otherwise the

new data is stored as the comparison data and returned to the caller.

15. The IPCryptR2 enters the Critical Error state if the Cryptographic Algorithm Test or Continuous Random Number Generator Test fails. An error indicator is output by turning the Alarm LED red while in the Critical Error state. The Critical Error state may be exited by powering the module off then on.
16. The IPCryptR2 enters the Signature Validation Failure state if the Firmware Integrity test or Firmware Load test fails. A status message indicating success is not output over the status interface to indicate the Firmware Integrity test or Firmware Load test failed. While in this state the module will wait to be programmed and will not perform any other operations.
17. If all power up self-tests pass, the Alarm LED output will be clear.
18. The IPCryptR2 does not perform any cryptographic functions while in an error state.
19. No special procedures are required to maintain physical security of the module while delivering to operators. During manufacturing, all of the firmware modules are signed by the Motorola Programmed Signature Private key and the module is loaded with the Motorola Programmed Signature Public key. The signature of the firmware is verified to ensure the integrity of the module when it is delivered to authorized operators. In addition, a default CO password is used to control access to the module during initialization.

5.2. Motorola Imposed Security Rules

1. The IPCryptR2 does not support multiple concurrent operators.
2. The module does not support the output of plaintext or encrypted secret or private keys.

6. Identification and Authentication Policy

The IPCryptR2 supports a User role and a Crypto-Officer role.

The Crypto-Officer role is authenticated by a digital signature during the Program Update service and a password for the remaining Crypto-Officer services. The Crypto-Officer password is initialized to a default value during manufacturing and is sent in encrypted form to the module for authentication. After authenticating, the Crypto-Officer password may be changed at any time. After a configurable number of consecutive invalid authentication attempts the KPK is zeroized, a new KPK is generated, all KEKs and TEKs are invalidated (key status is marked invalid), the password is reset to the factory default, and the module enters an error state that can only be cleared by power cycling the module.

The User role is authenticated by a 256-bit AES key for the Transfer Key Variable and OTAR services and the ECDH public / private key pair for the Negotiate IPSec Session, Encrypt, and Decrypt services.

Table 4: Roles and Authentication

Role	Authentication Type	Authentication Mechanism	Strength of Authentication
Crypto-Officer	Role-Based	ECDSA-384 digital signature (192 bits of encryption strength) for Program Update service	The probability of a successful random attempt is 1 in 2 ^ 192 or less than 1 in 6e+57. As the Program Update service requires more than one minute to complete, the random attempt success rate during a one minute period cannot be lowered to less than 1 in 100,000.
Crypto-Officer	Role-Based	14-32 character ASCII password for all Crypto-Officer services except Program Update	Since the minimum password length is 14 ASCII printable characters and there are 95 ASCII printable characters, the probability of a successful random attempt is 1 in 95 ^ 14 or 1 in 4,876,749,791,155,298,590,087,890,625. The module limits the number of authentication attempts in one minute to 60. The probability of a successful random attempt during a one-minute period is 60 in 95 ^ 14 or 1 in 81,279,163,185,921,643,168,131,510.
User (KVL Interface)	Role-Based	256-bit AES Black Keyloading Key	The probability of a successful random attempt is 1 in 2 ^ 256 or less than 1 in 1e+77.

Role	Authentication Type	Authentication Mechanism	Strength of Authentication
		for Transfer Key Variable service	Since it takes at least 1 sec per keyload, the probability of a successful random attempt during a one-minute period is 60 in 2 ^ 256 or less than 1 in 1.9e+75.
User (Ethernet Interface)	Role-Based	256-bit AES KEK for the OTAR service	The probability of a successful random attempt is 1 in 2 ^ 256 or less than 1 in 1e+77. Since it takes at least 1 sec per OTAR operation, the probability of a successful random attempt during a one-minute period is 60 in 2 ^ 256 or less than 1 in 1.9e+75.
User (Ethernet Interface)	Role-Based	256-bit TEK for the Negotiate IPSec Session, Encrypt, and Decrypt services	The probability of a successful random attempt is 1 in 2 ^ 256 or less than 1e+77. Since it takes at least 1 sec to negotiate an IPSec session, the probability of a successful random attempt during a one-minute period is 60 in 2 ^ 256 or less than 1 in 1.9e+75.

7. Physical Security Policy

The IPCryptR2 is a production grade, multi-chip standalone cryptographic module as defined by FIPS 140-2 and is designed to meet level 2 physical security requirements.

The IPCryptR2 is entirely contained within a hard plastic production-grade removable enclosure. The IPCryptR2 enclosure is opaque within the visible spectrum. The removable cover is protected with tamper-evident tape. The tamper-evident tape is visible on both side of the enclosure exterior.

Two (2) tamper-evident seals (see Figures 4 through 8) are installed during manufacturing and should be checked every six months by the user for signs of tamper.

No maintenance access interface is available.

Figure 4: Two (2) Tamper-Evident Seals From Underside (See Arrows)

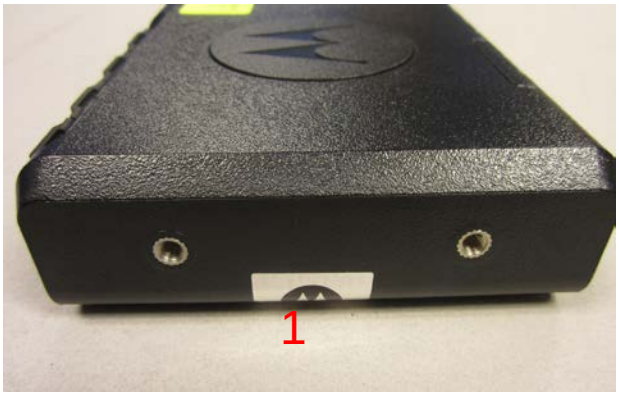

Figure 5: Tamper-Evident Seal #1 (Top Left Side)

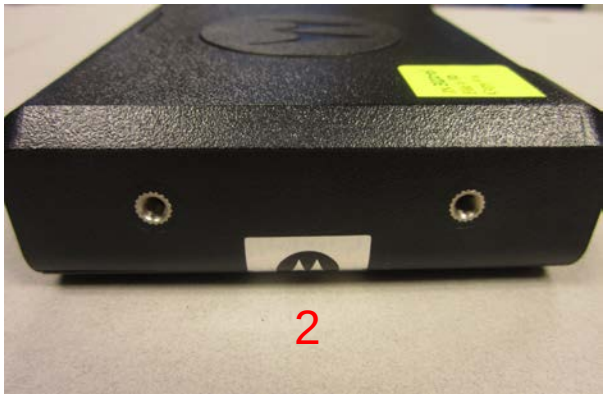

Figure 6: Tamper-Evident Seal #2 (Top Right Side)

Figure 7: Tamper-Evident Seal #1 (Underside Left Side)

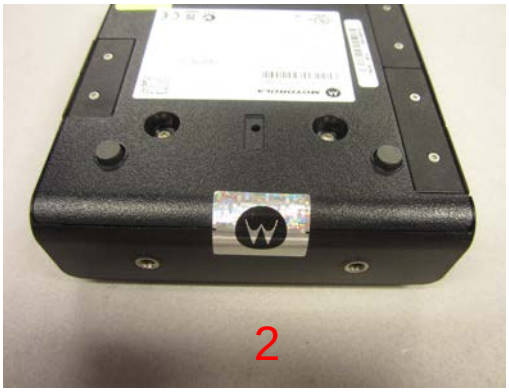

Figure 8: Tamper-Evident Seal #2 (Underside Right Side)

Tamper seals should be inspected along the entire seal's perimeter, its surface and the area immediately surrounding the seal for scratches, scrapes, gouges, cuts and any other signs of tampering. If such markings are found and/or the seals have been removed, the unit should be removed from service and returned to Motorola for proper servicing.

8. Access Control Policy

8.1. IPCryptR2 Supported Roles

The IPCryptR2 supports two (2) roles. These roles are defined to be the:

- User Role and,
- Crypto-Officer Role.

8.2. IPCryptR2 Services Available to the User Role.

- Transfer Key Variable: Transfer key variables (KEKs and TEKs) to the key database via the KVL interface.
- Key Check: Obtain status information about a specific KEK or TEK via the KVL interface.
- Invalidate Keys Via KVL: Invalidate KEKs and TEKs from the key database via the KVL interface.
- Negotiate IPSec session: establish an IPSec tunnel via the Ethernet port.
- Encrypt: Encrypt plaintext data received over the Ethernet port.
- Decrypt: Decrypt ciphertext data received over the Ethernet port.
- OTAR: Modify, query, and zeroize the Key Database via OTAR Key Management Messages.
- Generate Random Number: Generate random data using the Non-deterministic Hardware Random Number Generator and output result over serial shell interface. Available in both FIPS and non-FIPS mode.

8.3. IPCryptR2 Services Available to the Crypto-Officer Role.

- Program Update: Update the module firmware using TFTP. Firmware upgrades are authenticated and encrypted. Program Update will zeroize all keys and CSPs if an option to do so is selected via the serial shell.
- Validate Crypto-Officer Password: Validate the current Crypto-Officer password used to identify and authenticate the Crypto-Officer role via the shell. Successful authentication will allow entrance to the shell command interface and access to the shell command services.
- Change Crypto-Officer Password: Modify the current password used to identify and authenticate the Crypto-Officer Role via a shell command.
- Configure IPCryptR2: Set ISAKMP, IKE, and general configuration parameters via a shell command.
- Extract Error Log: Status request via a shell command. Provides detailed history of error events.
- Tunnel config: Provides the configuration for IKE via a shell command.
- Version Query (includes required Show Status service): Provides module firmware and hardware version numbers via a shell command.
- Shell Help: Shell command to get help on the format of other shell commands.
- Exit Shell: Exits the shell command interface and logs out of the Crypto-Officer role.
- Factory default: Performed using a shell command. Zeroizes TEK, KEK, ECDH Private / Public key pair, IPSec Session Keys, KPK and Crypto-Officer Password.

- Generate Key Pair: creates a key pair for an IKEv2 certificate-based tunnel
- Delete Key Pair: deletes a key pair generated from the “Generate Key Pair” service
- Generate CSR: generates a Certificate Signing Request (CSR) from a key pair generated by the “Generate Key Pair”
- Validate Certificate: validates a certificate chain for an IKEv2 certificate-based tunnel. This service also provides option to pick Motorola certificate extensions or customized certificate extensions.

8.4. IPCryptR2 Services Available Without a Role.

- DHCP: Used for automation of network parameters. This service does not access any CSPs or other security relevant functionality.
- Reset Crypto Module: Toggle the Reset input or a transition from power off to power on state.
- Perform Self-Tests: Performs module self-tests comprised of cryptographic algorithms test and firmware test. Initiated by module reset or transition from power off state to power on state.
- Zeroize All Keys: Invalidate KEKs and TEKs and zeroizes KPK, ECDH private/public key pair, and IPSec Session keys with Erase button.
- LED Status: LED output

8.5. Critical Security Parameters (CSPs) and Public Keys

Table 5: CSP Definition

CSP Identifier	Description
Black Keyloading Key (BKK)	256 bit AES Key used for decrypting keys entered into the module via a KVL. Stored in plaintext in non-volatile memory and zeroized through the Program Update service. The BKK is entered using the Program Update service and is not output from the module.
Image Decryption Key (IDK)	A 256-bit AES key used to decrypt downloaded images. Stored in plaintext in non-volatile memory and zeroized through the Program Update service. The IDK is entered using the Program Update service and is not output from the module.
Traffic Encryption Key (TEK)	256 bit IKE Pre-Shared Keys used for IKE authentication and OTAR. TEKs are entered in encrypted form via the KVL and via OTAR. TEKs entered via the KVL are wrapped with the BKK; TEKs received via OTAR are encrypted on a KEK. Stored in plaintext in RAM and encrypted by the KPK in flash. TEKs are not output from the module. Key is zeroized via OTAR, KVL, or by pressing the Erase Button.
Key Encryption Keys (KEKs)	256 bit AES Keys used for encryption of keys in OTAR. KEKs are entered in encrypted form via the KVL and via OTAR. KEKs entered via the KVL are wrapped with the BKK; KEKs received via OTAR are encrypted on another KEK. Stored in plaintext in RAM and encrypted by the KPK in flash. KEKs are not output from the module. Key is zeroized via OTAR, KVL, or by pressing the Erase Button.
Elliptic Curve Diffie-Hellman Private value	Randomly generated internally by IKE. Used in elliptic curve public-private key pair, to establish a shared secret over an insecure channel. Stored in volatile memory. The Elliptic Curve Diffie-Hellman Private value is not entered into or output from the module. Key is zeroized when the tunnel is destroyed or by resetting the module.
IPSec Session Keys	256 bit AES-GCM Key generated internally by IKE and used for data encryption. Stored in volatile memory. The IPSec Session Keys are not entered into or output from the module. Key is zeroized when the tunnel is destroyed or by resetting the module.
Key Protection Key (KPK)	256 bit AES key used to encrypt TEKs and KEKs for storage in non-volatile memory. Generated internally using Non-deterministic Hardware Random Number Generator. Stored in battery-backed RAM. The KPK is not entered into or output from the module. Key is zeroized when the module detects tamper or by pressing the Erase button.
Password Encryption Key (PEK)	This is a 256-bit AES Key used for decrypting passwords during password validation. Loaded via the Program Update

CSP Identifier	Description
	service. Stored in plaintext in non-volatile memory and zeroized through the Program Update service. Also stored encrypted with the KPK in non volatile memory. The PEK is not output from the module. Entry - on Program Update service request Output - n/a Storage - in plaintext in non volatile memory Zeroization - on Program Update service request Generation - n/a
Crypto-Officer Password	The Crypto-Officer password (14-32 ASCII printable characters in length) is entered encrypted on the PEK. After decryption the plaintext password is not stored but temporarily exists in volatile memory. The SHA-384 hash of the decrypted password is compared with the hash value stored in non-volatile memory during password validation. The Crypto-Officer Password is entered encrypted with the PEK and is not output from the module. Password is zeroized upon detecting the maximum number of invalid login attempts. During new password entry, the new password is decrypted with the PEK and hashed. It is subsequently stored encrypted with the KPK in non-volatile memory.
SNMP message encryption key	AES128 key for encrypting of SNMP messages, derived from an SNMP User or Admin password according to http://www.ietf.org/rfc/rfc3826.txt section 3.1.2.1. AES Encryption Key and IV. Key is in volatile memory. Key is zeroized upon power-up. The SNMP protocol utilized the following passwords which are stored in the clear: • Admin authentication password • Admin Privacy password • User authentication password • User Privacy password
IKEv2 integrity protection key	HMAC SHA-384 key as required by Suite B Cryptographic Suite for IPSEC. This key is used for IKEv2 Integrity protection. Key is zeroized when the tunnel is destroyed or by resetting the module.

Table 6: Public Keys

Key	Description
Public Programmed Signature Key	384 bit ECDSA key used to validate the signature of the firmware image being loaded before it is allowed to be executed. Stored in non-volatile memory. Loaded during manufacturing and as part of the boot image during a Program Update service. The Public Programmed Signature Key is not output from the module.

Elliptic Curve Diffie-Hellman Public value	Randomly generated Internally by IKE. Used in elliptic curve public-private key pair, to establish a shared secret over an insecure channel. Stored in volatile memory. The Elliptic Curve Diffie-Hellman Public value is generated internally and is output as part of the Diffie-Hellman key agreement protocol.

8.6. CSP Access Types

Table 7: CSP Access Types

CSP Access Type	Description
C – Check CSP	Checks status and key identifier information of key.
D – Decrypt CSP	Decrypts KEKs and TEKs retrieved from volatile memory using the KPK. Decrypts KEKs and TEKs entered via the KVL using the Black Keyloading Key. Decrypts KEKs entered via OTAR using other KEKs. Decrypts entered password with PEK during password validation.
E – Encrypt CSP	Encrypts KEKs and TEKs with KPK prior to storage in volatile memory.
G – Generate CSP	Generates KPK, IPSec Session keys, or Elliptic Curve Diffie-Hellman private key.
I – Invalidate CSP	Marks encrypted KEKs and TEKs stored in volatile memory as invalid. KEKs and TEKs marked invalid can then be over-written when new KEKs and/or TEKs are stored.
S – Store CSP	Stores KPK in volatile memory. Stores encrypted KEKs and TEKs in non-volatile memory, over-writing any previously invalidated KEK or TEK in that location. Stores plaintext Private/Public Elliptic Curve Diffie-Hellman values and IPSec Session Keys in volatile memory. Stores plaintext BKK, PEK, or IDK in non-volatile memory.
U – Use CSP	Uses CSP internally for encryption / decryption services.
Z – Zeroize CSP	Zeroizes CSP.

Table 8: CSP versus CSP Access

	CSP											Role		
Service	BKK (Black Keyloading Key)	IDK (Image Decryption Key)	TEK (Traffic Encryption Keys)	KEK (Key Encryption Keys)	ECDH Private / Public key pair	IPSec Session Keys	KPK (Key Protection Key)	PEK (Password Encryption Key	Crypto-Officer Password	SNMP mesg encryption key	IKEv2 integrity protection key	User Role	Crypto-Officer Role	No Role Required
1. Program Update	z, s	u, z, s	z[1]	z[1]	z	z	z	z,s					√	
2. Validate Crypto-Officer Password			i	i			z, g, s	u	d, u, z				√	
3. Change Crypto-Officer Password			i	i			z, g, s	u	d, u, z				√	
4. Configure IPCryptR2													√	
5. Extract Error Log													√	
6. Tunnel config											z		√	
7. Version Query													√	
8. Shell Help													√	
9. Exit Shell													√	
10. Factory default			z	z	z	z	z		z ,s	z	z		√	
11. Generate Key Pair					g,s								√	
12. Delete Key Pair					z								√	
13. Generate CSR					c,s, u								√	
14. Validate Certificate													√	
15. Transfer Key Variable	u		d, i, e, z, s	d, i, e, z, s			u					√		
16. Negotiate IPSec sessions					g, s, u	g, s					u	√		
17. Encrypt			d			u	u					√		
18. Decrypt			d			u	u					√		
19. OTAR	u		d, u, i, e, z, s	d, u, i, e, z, s			u					√		
20. Generate Random Number												√		

[1] Program Update will zeroize all keys and CSPs if an option to do so is selected via the serial shell

	CSP											Role		
Service	BKK (Black Keyloading Key)	IDK (Image Decryption Key)	TEK (Traffic Encryption Keys)	KEK (Key Encryption Keys)	ECDH Private / Public key pair	IPSec Session Keys	KPK (Key Protection Key)	PEK (Password Encryption Key	Crypto-Officer Password	SNMP mesg encryption key	IKEv2 integrity protection key	User Role	Crypto-Officer Role	No Role Required
21. DHCP												√	√	√
22. Key Check			c	c								√	√	
23. Reset Crypto Module							g, s			z	z	√	√	√
24. Perform Self-Tests												√	√	√
25. Invalidate Keys Via KVL			i	i								√	√	
26. Zeroize All Keys			i	i	z	z	z, g, s					√	√	√
27. LED Status												√	√	√

9. Mitigation of Other Attacks Policy

The IPCryptR2 is not designed to mitigate any specific attacks outside of those required by FIPS 140-2, including but not limited to power consumption, timing, fault induction, or TEMPEST attacks.