Security Policy: IPCryptR Motorola Advanced Crypto Engine (MACE)

Cryptographic module used in Motorola's IPCryptR for Astro, Dimetra and WiMAX systems

Version: R01.00.06

Date: April 14, 2010

Table of Contents

1. Introduction

1.1. Scope

This Security Policy specifies the security rules under which the IPCryptR Motorola Advanced Crypto Engine, herein identified as the IPCryptR MACE, must operate. Included in these rules are those derived from the security requirements of FIPS 140-2 and those imposed additionally by 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
CO	Crypto-Officer
CSP	Critical Security Parameter
DES	Data Encryption Standard
ECB	Electronic Code Book
ECDH	Elliptic Curve Diffie-Hellman
EI	Ethernet Interface
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
LFSR	Linear Feedback Shift Register
MACE	Motorola Advanced Crypto Engine
PEK	Password Encryption Key
PSK	Pre-Shared Keys
RAM	Random Access Memory
RNG	Random Number Generator

1.3. Overview

The IPCryptR MACE provides secure key management and data encryption for the IPCryptR in Astro, Dimetra and Broadband Systems.

1.4. IPCryptR MACE Implementation

The IPCryptR MACE is implemented as a single-chip cryptographic module as defined by FIPS 140-2.

1.5. IPCryptR MACE Hardware / Firmware Version Numbers

FIPS Validated Cryptographic Module Hardware Kit Numbers	FIPS Validated Cryptographic Module Firmware Version Numbers
5185912Y01	R01.01.02, R01.01.03

1.6. IPCryptR MACE Cryptographic Boundary

The IPCryptR MACE Cryptographic Boundary is drawn around the MACE IC as shown below.

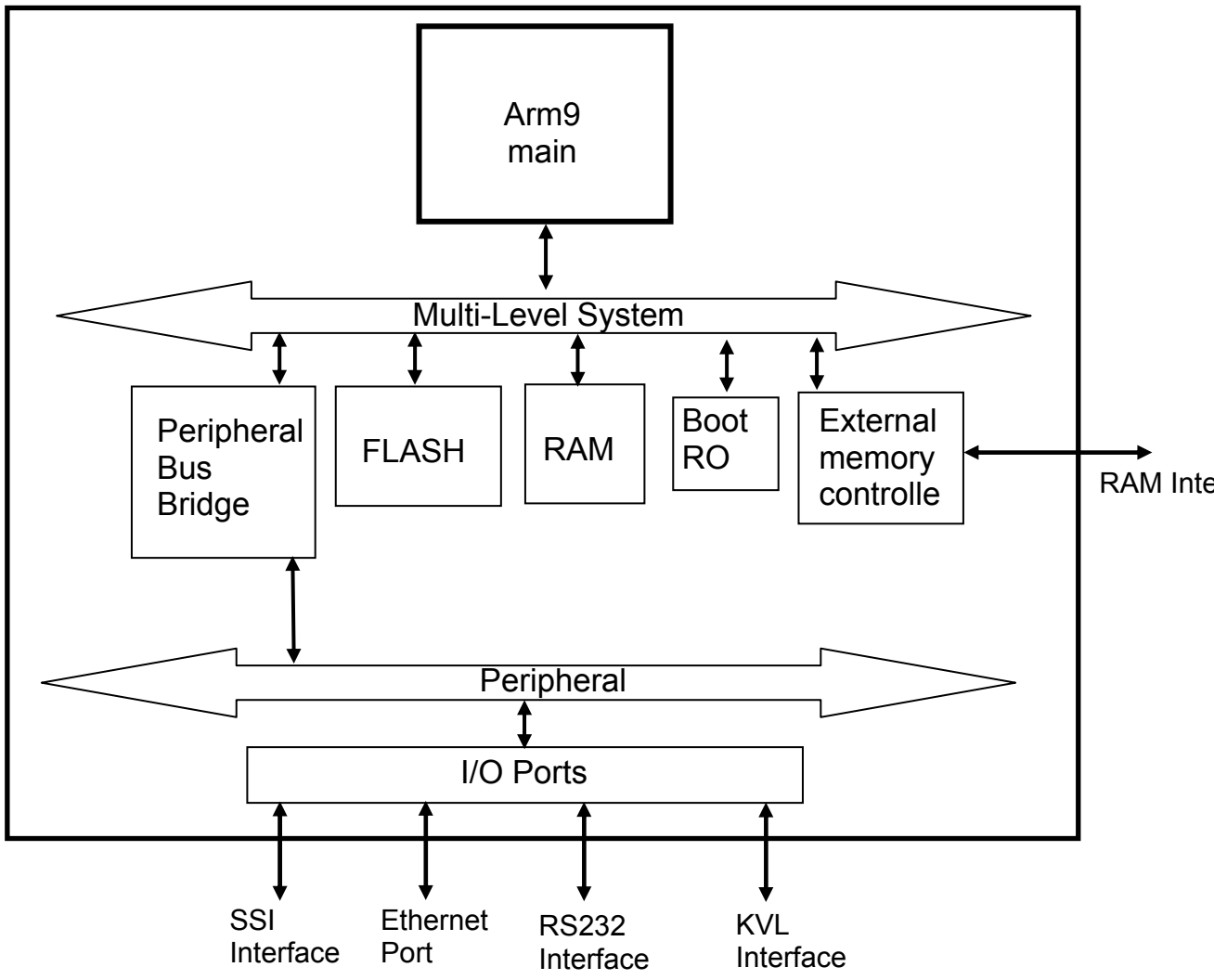

Figure 1: IPCryptR MACE Block Diagram

The Cryptographic Boundary is drawn around the IPCryptR MACE IC which is responsible for all key storage and generation and performs all crypto processing for the IPCryptR MACE.

1.7. Ports and Interfaces

The IPCryptR MACE provides the following physical ports and logical interfaces:

Table 1: Ports and Interfaces

Physical Port	Qty	Logical interface definition	Description
RS232 Interface	1	• Control Input • Status Output • Data Output	Provides an interface for factory programming and execution of RS232 shell commands. This interface does not support output of CSP's.
Serial Synchronous Interface (SSI)	1	• Data Input • Data Output • Control Input • Status Output	Provides an interface to the unprotected network and entry of the User password in encrypted form. This interface does not support output of CSP's.
Ethernet Port (EP)	1	• 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. Only the public key from the ECDH pair is output over the Ethernet port. This interface does not support any other input / output of CSP's.
Key Variable Loader (KVL)	1	• Data Input • Data Output • Control Input • Status Output	Provides an interface to the Key Variable Loader. The Pre-Shared Key (PSK) is entered in encrypted form over the KVL interface. This interface does not support output of CSP's.
RAM	1	• Data Input • Data Output • Control Input • Status Output	This interface provides storage for non-security related stack information. This interface does not support input / output of CSP's.
Power	1	• Power Input • Internal battery-backed RAM	This interface powers all circuitry. This interface does not support input / output of CSP's.

Physical Port	Qty	Logical interface definition	Description
Tamper Interface	1	• Control Input	The interface is used for zeroization of Pre-Shared Keys (PSK's), KPK, ECDH private/public key pair, and Ipsec Session keys.
Reset Interface	1	• Control Input	This interface forces a reset of the module.
Alarm LED output	1	• Status Output	The Alarm LED output turns red to indicate a fatal error has been detected.
Power LED output	1	• Status Output	The Power LED output turns green when power is supplied to the module.
Ready LED output	1	• Status Output	The Ready LED output turns green when the module is ready to communicate with a KVL.
TX Clear LED output	1	• Status Output	The TX Clear LED output is not used and remains off other than during power up self-test when the LED turns green.
Status LED output	1	• Status Output	The Status LED output turns green to indicate a good battery, a Pre-Shared Key (PSK) has been loaded, and a tunnel has been established. The Status LED output turns clear to indicate a good battery, a Pre-Shared Key (PSK) has been loaded, but a tunnel has not been established. The Status LED output turns yellow to indicate a good battery, but no Pre-Shared Key (PSK) has been loaded. The Status LED output turns red to indicate a low or dead battery.
IRQ/FIQ	2	• Control Input	External interrupts.
Clock	1	• Control Input	Clock input

2. FIPS 140-2 Security Levels

The IPCryptR MACE is designed to operate at FIPS 140-2 overall Security Level 3. 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 2: IPCryptR MACE Security Levels

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

3. FIPS 140-2 Approved Operational Modes

The IPCryptR MACE is designed to operate in a FIPS 140-2 Approved mode of operation at overall Security Level 3; the module does not support a non-FIPS Approved mode of operation.

The RS232 Interface Version Query command will be used to retrieve the current FW and HW version of IPCryptR MACE.

The module supports the following Approved algorithms:

- AES-256 (Cert. #819) – for symmetric encryption / decryption of keys and parameters stored in the internal database used in the following approved modes: CBC, and 8-bit CFB.
- AES-256 GCM (Cert. #1013) – for high-speed encryption and authentication in the GCM mode.
- SHA-256 (Cert. #817) – used for password hashing for internal password storage and digital signature verification during firmware integrity test and firmware load test
- SHA-384 (Cert. #963) – used as data origin authentication and integrity verification mechanisms for IKE.
- RSA-2048 (Cert. #396) – used for digital signature verification during firmware integrity test and firmware load test
- ANSI x9.31 RNG (Cert. #471) – used for IV and KPK generation

The module supports the following non-FIPS Approved algorithms:

- AES MAC (AES Cert. #819, vendor affirmed; P25 AES OTAR); AES (Cert. #819, key wrapping; key agreement methodology provides 256 bits of encryption strength)
- EC Diffie-Hellman – Asymmetric algorithms used for establishing secure private communication between two parties EC Diffie-Hellman (key agreement; key establishment methodology provides 192 bits of encryption strength)
- Maximal length 64-bit LFSR
- Non-deterministic Hardware Random Number Generator – used to provide random numbers used as Initialization Vectors (IV) and the seeds for the Approved RNG

4. Security Rules

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

4.1. FIPS 140-2 Imposed Security Rules

1. The IPCryptR MACE inhibits all data output via the data output interface whenever an error state exists and during self-tests.
2. The IPCryptR MACE logically disconnects the output data path from the circuitry and processes when performing key generation, manual key entry, or key zeroization.
3. Authentication data (e.g. passwords) are entered in encrypted form. Authentication data is not output during entry.
4. Secret cryptographic keys are entered in encrypted form over a physically separate port.
5. The IPCryptR MACE enforces Identity-Based authentication.
6. The IPCryptR MACE supports a User role and a Cryptographic Officer role. Authenticated operators are authorized to assume either supported role. The module does not allow the operator to change roles.
7. The IPCryptR MACE re-authenticates an operator when it is powered-up after being powered-off.
8. The IPCryptR MACE prevents brute-force attacks on its Cryptographic Officer password by using a password that is a minimum of 8 and a maximum of 16 ASCII printable characters in length. The probability of a successful random attempt is at least 1 in 63,527,879,748,485,376. It would require at least 635,278,797,484 attempts in one minute to lower the random attempt success rate to less than 1 in 100,000. A limit of 10 failed authentication attempts is imposed: 10 consecutive failed authentication attempts will cause the KPK to be zeroized, a new KPK to be generated, all PSK's to be invalidated (key status is marked invalid), the password to be reset to the factory default, and the module to enter an error state.
9. The IPCryptR MACE prevents brute-force attacks on its User password by using a password that is 10 hexadecimal digits long. The probability of a successful random attempt is 1 in 1,099,511,627,776. It would require 10,995,116 attempts in one minute to lower the random attempt success rate to less than 1 in 100,000. A limit of 15 failed authentication attempts is imposed: 15 consecutive failed authentication attempts will cause the KPK to be zeroized, a new KPK to be generated, and all PSK's to be invalidated (key status is marked invalid).
10. The IPCryptR MACE uses RSA-2048 to prevent brute-force attacks on the digital signature used to verify firmware integrity during a Program Update. 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.
11. Authentication data is not output during entry.
12. The IPCryptR MACE implements all firmware using a high-level language, except the limited use of low-level languages to enhance performance.
13. The IPCryptR MACE protects secret keys and private keys from unauthorized

disclosure, modification and substitution.

14. The IPCryptR MACE provides a means to ensure that a key entered into or stored within the IPCryptR MACE is associated with the correct entities to which the key is assigned. Each key in the IPCryptR MACE 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.
15. The IPCryptR MACE denies access to plaintext secret and private keys contained within the IPCryptR MACE.
16. The IPCryptR MACE provides the capability to zeroize all plaintext cryptographic keys and other unprotected critical security parameters within the module.
17. The IPCryptR MACE conforms to FCC 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B requirements.
18. The IPCryptR MACE 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-256, SHA-384, and AES-256 in GCM, 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.
      - RNG KAT test: the RNG is initialized with a known answer seed, DT counter and Triple-DES key. The RNG is run and the result compared to known answer data. The test passes if the generated data matches the known answer data, otherwise the test fails.
      - Firmware integrity test: A digital signature is generated over the code when it is built using SHA-256 and RSA-2048 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 the test passes, otherwise it fails.
      - External indicators test: Upon every power up, the MACE 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 MACE.
    - Conditional tests
      - Firmware load test: A digital signature is generated over the code when it is built using SHA-256 and RSA-2048. Upon download into the module, the

digital signature is verified. If the digital signature matches the test passes, otherwise it fails.

- Continuous Random Number Generator test: The continuous random number generator test is performed on all RNGs supported by the module. For each RNG, 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 any one of the RNGs 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.

19. The IPCryptR MACE enters an error state if the Cryptographic Algorithm Test, Continuous Random Number Generator Test, or RNG KAT fails. This error state may be exited by powering the module off then on.
20. The IPCryptR MACE enters an error state if the Firmware Integrity test or Firmware Load test fails. As soon as an error indicator is output via the status interface, the module transitions from the error state to a state that only allows new firmware to be loaded.
21. The IPCryptR MACE outputs an error indicator by turning the Alarm LED output red whenever an error state is entered due to a failed self-test. If all power up self-tests pass, the Alarm LED output will be clear.
22. The IPCryptR MACE does not perform any cryptographic functions while in an error state.

4.2. Motorola Imposed Security Rules

1. The IPCryptR MACE does not support multiple concurrent operators.
2. All cryptographic module services are suspended during key loading.
3. After a sufficient number (10) of consecutive unsuccessful user login attempts, the module will zeroize all keys from the Key Database.
4. The module does not support the output of plaintext or encrypted secret or private keys.

5. Identification and Authentication Policy

The IPCryptR MACE 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 which is a minimum of 8 and maximum of 16 ASCII printable characters in length for the remaining Crypto-Officer services. After authenticating, the CO password may be changed at any time. After ten consecutive invalid authentication attempts the KPK is zeroized, a new KPK is generated, all PSK's 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.

A 10-digit hexadecimal password is used to authenticate the User role. The User password cannot be changed. After fifteen consecutive invalid authentication attempts the KPK is zeroized, a new KPK is generated, and all PSK's are invalidated (key status is marked invalid).

Both Crypto-Officer and User ID's and passwords are initialized to a default value during manufacturing and are sent in encrypted form to the MACE for authentication.

Role	Authentication Type	Identification	Authentication Data Required
Crypto-Officer	Identity-Based	Crypto-Officer ID	Digital signature for Program Update service 8-16 character ASCII password
User	Identity-Based	User ID	10-digit hexadecimal password

6. Physical Security Policy

The IPCryptR MACE is a production grade, single-chip cryptographic module as defined by FIPS 140-2 and is designed to meet level 3 physical security requirements.

The IPCryptR MACE is covered with a hard opaque metallic coating that provides evidence of attempts to tamper with the module. Tampering with the module will cause it to enter a lock-up state in which no crypto services will be available. The IPCryptR MACE does not contain any doors, removable covers, or ventilation holes or slits. No maintenance access interface is available.

7. Access Control Policy

7.1. IPCryptR MACE Supported Roles

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

- User Role and,
- Crypto-Officer (CO) Role.

The IPCryptR MACE supports only one User ID and one Crypto-Officer ID.

7.2. IPCryptR MACE Services Available to the User Role.

- Transfer Key Variable: Transfer key variables (PSK) to the MACE key database via the KVL interface.
- Key Check: Obtain status information about a specific PSK via the KVL interface.
- Validate User Password: Validate the current User password used to identify and authenticate the User role via the SSI interface. Fifteen consecutive failed validation attempts will cause the KPK to be zeroized, a new KPK to be generated, and the PSK to be invalidated (key status is marked invalid).
- Zeroize Keys Via KVL: Zeroize PSKs from the key katabase via the KVL interface.
- Negotiate IPsec session: establish an IPsec tunnel via the Ethernet port.
- Encrypt: Encrypt plaintext data received over the Ethernet port and return ciphertext over SSI.
- Decrypt: Decrypt ciphertext data received over the SSI and send plaintext over Ethernet port.

7.3. IPCryptR MACE Services Available to the Crypto-Officer Role.

- Program Update: Update the module firmware via the KVL interface. Firmware upgrades are authenticated using a digital signature. All keys and CSPs are zeroized during a Program Update.
- Validate Crypto-Officer Password: Validate the current Crypto-Officer password used to identify and authenticate the Crypto-Officer role via the RS232 interface. Successful authentication will allow entrance to the RS232 shell command interface and access to the RS232 shell command services. Ten consecutive failed attempts causes the KPK to be zeroized, a new KPK to be generated, the PSK to be invalidated (key status is marked invalid), the password to be reset to the factory default, and the module to enter an error state that can only be cleared by power cycling the module.
- Change Crypto-Officer Password: Modify the current password used to identify and authenticate the CO Role via an RS232 shell command. Ten consecutive failed attempts causes the KPK to be zeroized, a new KPK to be generated, PSK to be invalidated (key status is marked invalid), the password to be reset to the factory default, and the module to enter an error state that can only be cleared by power cycling the module.
- Configure IPCryptR: Set configuration parameters used in the ISAKMP and IKE protocols via an RS232 shell command.
- Extract Error Log: Status request via an RS232 shell command. Provides detailed history of error events.
- Tunnel config: Provides the configuration for IKE via an RS232 shell command.

- Version Query: Provides module firmware and hardware version numbers via an RS232 shell command.
- RS232 Shell Help: Shell command to get help on the format of other RS232 shell commands.
- Exit RS232 Shell: Exits the RS232 shell command interface and logs out of the Crypto-Officer role.

7.4. IPCryptR MACE Services Available Without a Role.

- Reset Crypto Module: Toggle the Reset input or a transition from power off to power on state.
- Initiate 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: Zeroizes the PSK, KPK, ECDH private/public key pair, and Ipsec Session keys via the Tamper interface.

7.5. Critical Security Parameters (CSPs) and Public Keys

Table 3: CSP Definition

CSP Identifier	Description
ANSI X9.31 seed	A 64-bit seed value used within the ANSI X9.31 RNG. The seed is not stored but temporarily exists in volatile memory and is zeroized by power cycling the module. The seed is not entered into or output from the module.
ANSI X9.31 seed key	This is a 128 bit TDES Key used to seed the ANSI X9.31 RNG during initialization. The seed key is not stored but temporarily exists in volatile memory and is zeroized by power cycling the module. The seed key is not entered into or output from the module.
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.
Pre-Shared Key (PSK)	256 bit AES Key used for IKE authentication. The PSK is entered in encrypted form via the KVL. Stored in plaintext in RAM and encrypted by the KPK in flash. The PSK is entered wrapped with the BKK and is not output from the module.
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.
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 Protection Key (KPK)	256 bit AES key used to encrypt PSKs. Generated internally by the ANSI X9.31 RNG. Stored in battery-backed RAM. The KPK is not entered into or output from the module.
Password Encryption Key (PEK)	256 bit AES Key used for decrypting passwords during password validation. Loaded via the Program Update service. Stored in plaintext in non-volatile memory and zeroized through the Program Update service. The PEK is entered using the Program Update service and is not output from the module.
User Password	The User password (10 hex digits in length) is entered encrypted on the PEK. After decryption the plaintext password is not stored but temporarily exists in volatile memory. The SHA-256 hash of the decrypted password is compared with the hash value stored in non-volatile memory during password validation. The User Password is entered encrypted with the PEK and is not output from the module.
Crypto-Officer Password	The Crypto-Officer password (8-16 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-256 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.

Table 4: Public Keys

Key	Description
Public Programmed Signature Key	2048 bit RSA key used to validate the signature of the firmware image being loaded before it is allowed to be executed. Stored in volatile memory. Loaded during manufacturing and as part of the boot image during a Program Update service. The Public Programmed Signature Key is loaded during manufacturing and 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.

7.6. CSP Access Types

Table 5: CSP Access Types

CSP Access Type	Description
c – Check CSP	Checks status and key identifier information of key.
D – Decrypt CSP	Decrypts PSK retrieved from volatile memory using the KPK. Decrypts PSKs entered via the KVL using the Black Keyloading Key. Decrypts entered password with PEK during password validation.
E – Encrypt CSP	Encrypts PSK with KPK prior to storage in volatile memory.
G – Generate CSP	Generates KPK, ANSI X9.31 seed, ANSI X9.31 seed key, or Elliptic Curve Diffie-Hellman private key.
I – Invalidate CSP	Marks encrypted PSKs stored in volatile memory as invalid. PSKs marked invalid can then be over-written when new PSKs are stored.
S – Store CSP	Stores KPK in volatile and volatile memory. Stores encrypted PSKs in non-volatile memory, over-writing any previously invalidated PSK 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 key.

Table 6: CSP versus CSP Access

	CSP											Role		
Service	ANSI X9.31 seed	ANSI X9.31 seed key	PSK (Pre-Shared Keys)	ECDH Private / Public key pair	Ipsec Session Keys	KPK (Key Protection Key)	PEK (Password Encryption Key	BKK (Black Keyloading Key)	IDK (Image Decryption Key)	User Password	Crypto-Officer Password	User Role	Crypto-Officer Role	No Role Required
1. Program Update			z	z	z	z	z,s	z, s	u, z, s				√	
2. Validate Crypto-Officer Password			i			z, g, s	u				d, u, z		√	
3. Change Crypto-Officer Password			i			z, g, s	u				d, u, z		√	
4. Configure IPCryptR													√	
5. Extract Error Log													√	
6. Tunnel config													√	
7. Version Query													√	
8. RS232 Shell Help													√	
9. Exit RS232 Shell													√	
10. Transfer Key Variable			i, e, z, s			u		u				√		
11. Key Check			c									√		
12. Validate User Password			i			z, g, s	u			d, u, z		√		
13. Zeroize Keys Via KVL			i									√		
14. Negotiate IPsec sessions				g, s, u	g, s							√		
15. Encrypt			d		u	u						√		
16. Decrypt			d		u	u						√		
17. Reset Crypto Module	g, u, z	g, u, z				g, s						√	√	√
18. Initiate Self-Tests												√	√	√
19. Zeroize All Keys			i	z	z	z, g, s						√	√	√

8. Mitigation of Other Attacks Policy

The IPCryptR MACE 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.