Non-Proprietary FIPS 140-2 Security Policy:

µMACE

Cryptographic module for the Motorola Solutions CRYPTR Micro

Version: 1.5

Date: January 10, 2020

Table of Contents

List of Tables

List of Figures

1 Introduction

This document defines the Security Policy for the Motorola Solutions µMACE cryptographic module, hereafter denoted the Module. The Module provides secure key management, and voice/data encryption for the Motorola Solutions CRYPTR Micro.

Table 1 – Cryptographic Module Configuration

Module	HW P/N and Version	Base FW Version
Motorola µMACE	51009730001, Rev 0x0001	R03.01.12

Algorithms may also optionally be loaded into, or "Drop-in" the Module independent of the Base FW via the Program Update service.

Table 2 – Approved Mode Drop-in Algorithms

Algorithm	Algorithm FW Version	Cert. #
AES128	R01.00.01 (0x52010001)	5076
AES256	R01.00.03 (0x52010003)	5077

Table 3 – Non-Approved Mode Drop-in Algorithms

Algorithm	Algorithm FW Version
ADP	R01.00.00 (0x52010000)
DES-XL	R01.00.00 (0x52010000)
DES-OFB	R01.00.00 (0x52010000)
DES-ECB	R01.00.00 (0x52010000)
DES-CBC	R01.00.00 (0x52010000)
DVI-XL	R01.00.00 (0x52010000)
DVP-XL	R01.00.00 (0x52010000)
Localized Capable	R01.00.00 (0x52010000)

The Module is intended for use by markets that require FIPS 140-2 validated overall level 2 crypto-processors.

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

Table 4 – Security Level of Security Requirements

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

1.1 Module Description and Cryptographic Boundary

The physical form of the Module is depicted in Figure 1. The Module is a single chip embodiment. The cryptographic boundary is the surface and edges of the chip as shown in Figure 1 and Figure 2 below.

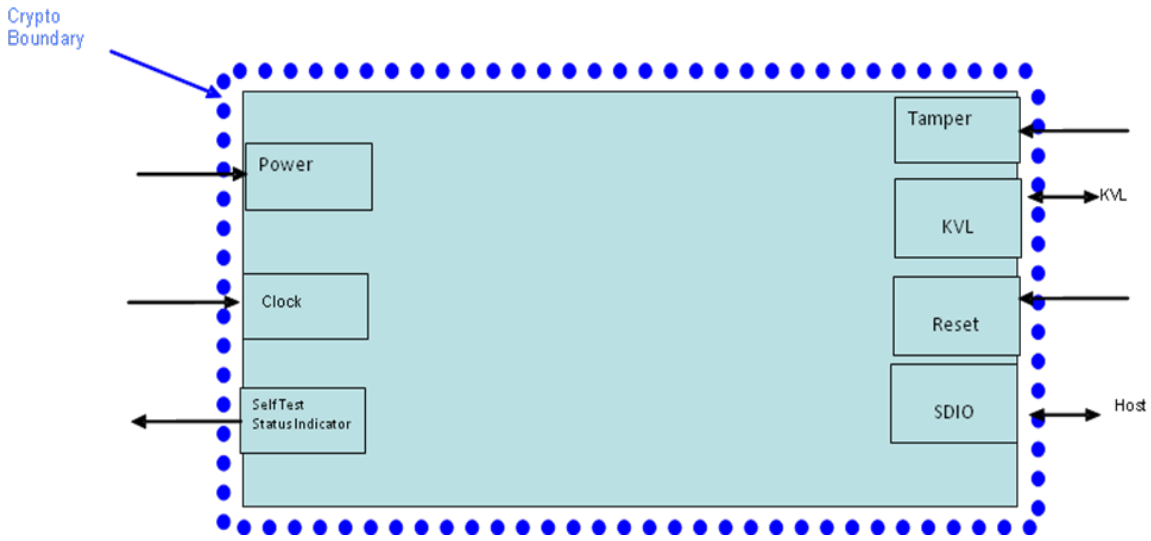

Figure 1 – Module Block Diagram

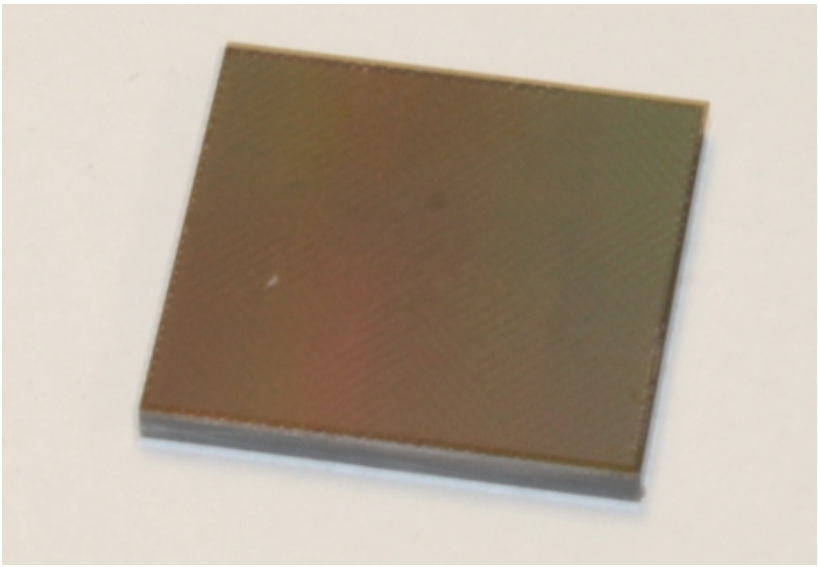

Figure 2 – Module

The Module's ports and associated FIPS defined logical interface categories are listed in Table 4.

Table 5 – Ports and Interfaces

Port	Description	Logical Interface Type
Power	This interface powers all circuitry. This interface does not support input / output of CSP's.	Power Input
Clock	Clock Input. Not used. This interface does not support input / output of CSP's.	Control Input
Tamper	Tamper Input. Not used. This interface does not support input / output of CSP's.	Control Input
KVL	SDIO DAT1 pin exclusively utilized by the Motorola Key Variable Loader (KVL). The KEKs and TEKs are entered in encrypted form over the KVL interface. All CSPs exchanged over this interface are always encrypted when operating in FIPS Approved mode.	Control Input Status Output Data Output Data Input
Self-test Status Indicator	This interface provides status output to indicate all power-up self-tests completed successfully.	Status Output
Reset	This interface forces a reset of the module. Not used.	Control Input
SDIO Interface	Provides an interface for factory programming and execution of SDIO commands. All CSPs exchanged over this interface are always encrypted when operating in FIPS Approved mode.	Control Input Status Output Data Output Data Input

2 Modes of Operation

The Module can be configured to operate in a FIPS 140-2 Approved mode of operation and a non-Approved mode of operation. To transition between FIPS 140-2 Approved and non-Approved modes, an operator must change the value of CSPs via the Program Update service as mentioned in section 3.1; all other CSPs are automatically zeroized by the Module when switching modes. To verify that the Module is in the Approved mode of operation, output from the Version Query service can be used as specified in Table 5. Note that AES-128 and AES-256 drop-in algorithms may or may not be loaded into the Module, however if they are loaded then the output of the Version Query service must also match the values in Table 2 to be in the Approved mode.

Table 6 – Approved Mode Indicator

Item ID	Value	Meaning
0x06 (FIPS)	0x03	FIPS Approved mode
0x06 (FIPS)	0x00	non-Approved mode

The Version Query service can also be used to verify the firmware version matches an approved version listed on NIST's website: http://csrc.nist.gov/groups/STM/cmvp/validation.html

2.1 Approved Mode Configuration

In order to configure the Module into an Approved mode, the Module Configuration service must be used to ensure the following parameters are disabled. Enabling any of the following parameters forces the Module to transition into a non-Approved mode. An operator must also adhere to the procedural enforcement requirements documented in Section 9.2 of this Security Policy.

1. Clear Key Import
2. Clear Key Export
3. Key Loss Key (KLK)
4. Red Keyloading

Additionally, the Module supports "drop-in algorithms" via the Program Update service. Drop-in algorithms may be added or removed from the Module independent of the base FW. In order to remain in the Approved mode, only Approved algorithms may be loaded into the Module; in particular AES-128 (Cert. #5076) and/ or AES-256 (Cert. #5077).

3 Cryptographic Functionality

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

Table 7 – Approved Algorithms

Cert	Algorithm	Mode	Description	Functions/Caveats
5075	AES [197]	ECB [38A]	Key Sizes: 128	Encrypt, Decrypt
5079	AES [197]	ECB [38A]	Key Sizes: 256	Encrypt, Decrypt
		GCM [38D]	Key Sizes: 256 Tag Len: 128	Authenticated Encrypt, Authenticated Decrypt
5076	AES [197]	CBC [38A]	Key Sizes: 128	Encrypt, Decrypt
		CTR [38A]	Key Sizes: 128	Encrypt, Decrypt
		OFB [38A]	Key Sizes: 128	Encrypt, Decrypt
5077	AES [197]	CBC [38A]	Key Sizes: 256	Encrypt, Decrypt
		CFB [38A]	Key Sizes: 256	Encrypt, Decrypt
		CTR [38A]	Key Sizes: 256	Encrypt, Decrypt
		OFB [38A]	Key Sizes: 256	Encrypt, Decrypt
5078	AES [197]	KW [38F]	Forward Key Sizes: 128, 256	Authenticated Encrypt, Authenticated Decrypt
VA	CKG [IG D.12]	[133] Section 6.1 Asymmetric signature key generation using unmodified NDRNG output		Key Generation
		[133] Section 6.2 Asymmetric key establishment key generation using unmodified NDRNG output		
		[133] Section 7.1 Direct symmetric key generation using unmodified NDRNG output		
		[133] Section 7.3 Derivation of symmetric keys from a key agreement shared secret.		
		[133] Section 7.4 Derivation of symmetric keys		

Cert	Algorithm	Mode	Description	Functions/Caveats
		from a pre-shared key		
1636	CVL: TLS [135]	v1.2	SHA(384)	Key Derivation
	CVL: SRTP [135]		AES-256	
C216	ECDSA [186]		P-384	KeyGen
			P-384 SHA(384)	SigGen
			P-384 SHA(384)	SigVer
3386	HMAC [198]	SHA-384	Key Sizes: 32 λ = *48*	Message Authentication, KDF Primitive, Password Obfuscation
3387	HMAC [198]	SHA-384	Key Sizes: 32 λ = *48*	Message Authentication, KDF Primitive, Password Obfuscation
C216	KAS [56A]	ECC (Initiator, Responder), KPG, Partial	P-384 SHA-384	Key Agreement Scheme provides 192 bits of encryption strength
N/A	KTS [38F]	KW, GCM	AES Cert. #5078	Key establishment methodology provides 128 or 256 bits of encryption strength
N/A	KTS [38F]	CBC, ECDSA	AES Cert. #5077 and ECDSA Cert. # C216	Key establishment methodology provides 256 bits of encryption strength
4132	SHS [180]	SHA-256 SHA-384		Message Digest Generation, Password Obfuscation
4133	SHS [180]	SHA-256 SHA-384		Message Digest Generation, Password Obfuscation

Table 8 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Description
AES128 Key Unwrap	[IG D.9] AES-OFB (Certs. #5076) key unwrapping for use in key transport; key establishment methodology provides 128 bits of encryption strength.
AES256 Key Unwrap	[IG D.9] AES-OFB (Certs. #5077) key unwrapping for use in key transport; key establishment methodology provides 256 bits of encryption strength.

Algorithm	Description
AES MAC	[IG G.13] AES MAC for Project 25 APCO OTAR (Cert. #5076 and #5077)
NDRNG	[IG G.13] Non-Deterministic RNG; 32 bits per access.

Table 9 – Non-Approved Cryptographic Functions

Algorithm	Description
AES Key Wrap	AES-OFB key wrapping for use in key transport.
DIA	Any drop-in algorithm (DIA) other than AES Certs. #5076 and #5077, as described in Section 2.1 of this Security Policy.

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

- ADP
- DES-XL
- DES-OFB
- DES-ECB
- DES-CBC
- DVI-XL
- DVP-XL
- Localized Capable
- AES-OFB Key Wrap

Note that all of the above are “drop-in” algorithms, except AES-OFB Key Wrap.

3.1 Critical Security Parameters

All CSPs used by the Module are described in this section. Usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in Section 4. It should be noted that Keys/CSPs stored in non-volatile memory/storage are normally preserved during a Program Update. However, all keys/CSPs are zeroized during a Program Update if one or more of the following occurs:

- The key database format/version changes between the resident and upgrade software images.
- The module’s FIPS status changes, post-upgrade (this indicates that a non-FIPS compliant Drop-in algorithm has been loaded onto the Module)

Table 10 – Critical Security Parameters (CSPs)

CSP	Description / Usage
Key Protection Key (KPK)	256-bit AES-CFB8 key used to encrypt TEKs, KEKs, the ECDSA Private Generated Signature Key, and the SRTP/ SRTCP Master Key stored in non-volatile memory.

CSP	Description / Usage
	• Entry: N/A • Output: N/A • Storage: Encrypted by the UKPPK in non-volatile memory • Zeroization: Program Update • Generation: NDRNG
Universal Key Protection Protection Key (UKPPK)	A 256-bit AES-OFB key used for encrypting the KPK. • Entry: Encrypted by the Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service • Output: N/A • Storage: Plaintext in volatile memory, plaintext in non-volatile memory • Zeroized: Program Update • Generation: N/A
Key Variable Loader Black Keyloading Key (KVL-BKK)	256-bit AES-OFB key used for encrypting keys that are input into the Module via the KVL interface. • Entry: Encrypted by the Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service • Output: N/A • Storage: Plaintext in volatile memory, plaintext in non-volatile memory • Zeroized: Program Update • Generation: N/A
Black Keyloading Key (BKK)	256-bit AES-OFB key used for encrypting keys that are input into the Module and output from the Module via the SDIO interface. • Entry: Encrypted by the Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service • Output: N/A • Storage: Plaintext in volatile memory, plaintext in non-volatile. • Zeroization: Program Update • Generation: N/A
Image Decryption Key (IDK)	A 256-bit AES-CBC key used to decrypt downloaded images. • Entry: Encrypted by the previous Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service • Output: N/A • Storage: Plaintext in volatile memory, plaintext split-knowledge in non-volatile memory • Zeroization: Program Update • Generation: N/A
Traffic Encryption Keys (TEKs)	128 and 256-bit AES-OFB keys used for enabling secure communication with target devices. These keys could also be used for HMAC Key, encryption and authentication of Key Management Messages in OTAR. • Entry: Encrypted by the KVL-BKK with AES256 OFB key unwrap when authenticated to the KVL role. Encrypted by either the BKK or a KEK with AES

CSP	Description / Usage
	SP 800-38F KTS, depending on host selection when authenticated to the User role. • Output: Encrypted by a KEK with AES SP 800-38F KTS (KW or OTAR format) over the SDIO Interface • Storage: Stored plaintext in volatile memory, encrypted by the KPK in non-volatile memory • Zeroization: Delete Key Variable and Program Update • Generation: Established through SP800-56A KAS
Key Encryption Keys (KEKs)	128 and 256-bit AES-KW or AES-OFB keys used for encryption of keys in key transport operation. It could be also used as an HMAC Key. • Entry: Encrypted by the KVL-BKK with AES256 OFB key unwrap when authenticated to the KVL role. Encrypted by either the BKK or a KEK with AES SP 80038F KTS, depending on host selection when authenticated to the User role. • Output: Encrypted by a KEK with AES SP 800-38F KTS (KW or OTAR format) over the SDIO Interface • Storage: Stored plaintext in volatile memory, encrypted by the KPK in non-volatile memory • Zeroization: Delete Key Variable and Program Update • Generation: Established through SP 800-56A KAS
Password Encryption Key (PEK)	256-bit AES-CFB8 key used for decrypting passwords during password validation. • Entry: Encrypted by the Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service • Output: N/A • Storage: Plaintext in non-volatile memory • Zeroization: Program Update • Generation: N/A
User Password	14-32 ASCII printable characters User authentication password. The SHA-384 hash of the decrypted password is compared with the SHA-384 hash value stored in non-volatile memory during password validation • Entry: Encrypted by the PEK with AES256-CFB8. • Output: N/A • Storage: Plaintext in volatile memory, SHA-384 hash of the plaintext password is encrypted by the PEK in non-volatile memory • Zeroization: Program Update, Change User Password • Generation: N/A
Crypto-Officer Password	14-32 ASCII printable characters Crypto-Officer authentication password. The SHA-384 hash value of the plaintext password is stored encrypted on the PEK in non-volatile memory. The SHA-384 hash of the decrypted password is compared with the SHA-384 hash value stored in non-volatile memory during password validation. • Entry: Encrypted by the PEK with AES256-CFB8. • Output: N/A

CSP	Description / Usage
	• Storage: Plaintext in volatile memory, SHA-384 hash of the plaintext password is encrypted by the PEK in non-volatile memory • Zeroization: Program Update, Change Crypto-Officer Password • Generation: N/A
Elliptic Curve Diffie-Hellman Private Key	Random value used to establish a shared secret over an insecure channel. • Entry: N/A • Output: N/A • Storage: Plaintext in volatile memory. • Zeroization: Delete Key Variable, Power cycle • Generation: FIPS 186-4 Key Generation on Perform Key Agreement Process service request
Elliptic Curve Diffie-Hellman Shared Secret	The Elliptic Curve Diffie-Hellman Shared Secret is output as part of the Diffie-Hellman key agreement protocol. • Entry - N/A • Output – N/A • Storage – Plaintext in volatile memory • Zeroization - Power cycle, Program Update • Generation - Established through SP800-56A KAS
ECDSA Private Generated Signature Key (PGSK)	384-bit ECDSA key used to generate the signature of the input data from the Generate Signature service request. • Entry: N/A • Output: N/A • Storage: Plaintext in volatile memory, encrypted by a KPK in non-volatile memory. • Zeroization: Delete Key Variable, Program Update, Power cycle • Generation: FIPS 186-4 Key Generation on Generate Key Variable service request
SRTP/SRTCP Master Key	256-bit master key used in the SRTP/SRTCP based derivation of KDF Derived Keys • Entry: Encrypted by the KVL-BKK with AES OFB key unwrap when authenticated to the KVL role. Encrypted by either the BKK or a KEK with AES SP 800-38F KTS, depending on host selection when authenticated to the User role. • Output: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW format) over the SDIO Interface • Storage: Stored plaintext in volatile memory, encrypted by the KPK in non-volatile memory • Zeroization: Delete Key Variable, Power cycle, Program Update • Generation: Internally generated using NDRNG or derived from SRTP/ SRTCP KDF on Generate Key Variable service request
SRTP/SRTCP Master Salt	112-bit key used to generate keys using SRTP KDF protocol, or 96-bit key to generate IV internally for AES GCM encryption operation. • Entry: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW or

CSP	Description / Usage
	OTAR format) or AES OFB key unwrap over the SDIO Interface • Output: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW format) over the SDIO Interface • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: Internally generated using NDRNG or derived from SRTP/ SRTCP KDF on Generate Key Variable service request
TLS KDF Secret	Secret input used in the TLS-based derivation of KDF Derived Keys. In practice, this input will typically be the Premaster Secret or Master Secret as defined in RFC 5246, but is dependent on the operator. • Entry: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW or OTAR format) or AES OFB key unwrap over the SDIO Interface • Output: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW format) over the SDIO Interface • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: Internally generated using NDRNG or derived from TLS KDF on Generate Key Variable service request
KDF Derived Key	Keys derived using TLS or SRTP KDFs. Module does not have control over the usage of these generated keys, the operator decides the usage. KDF output is always 384 bits, but a key of less length may be derived using a subset of this output. • Entry: N/A • Output: Encrypted by a KEK (User Role only) with AES SP 800-38F KTS (KW format) over the SDIO Interface • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: Internally derived through TLS or SRTP/SRTCP KDF on Generate Key Variable service request

3.2 Public Keys

Table 11 – Public Keys

Key	Description / Usage
ECDSA Public Programmed Signature Key	384-bit ECDSA public key used to validate the signature of the firmware image being loaded before it is allowed to be executed. • Entry: Encrypted by the Image Decryption Key and authenticated with the ECDSA Public Programmed Signature Key via the Program Update service. The first key is loaded in manufacturing. • Output: N/A • Storage: Plaintext in non-volatile memory • Zeroization: Program Update • Generation: N/A

Key	Description / Usage
ECDSA Public Generated Signature Key	384-bit ECDSA key used to verify signatures. • Entry: N/A • Output: Plaintext over the SDIO interface • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: FIPS 186-4 Key Generation on Generate Key Variable service request
Elliptic Curve Diffie-Hellman Public Key	Used to establish a shared secret over an insecure channel. • Entry: N/A • Output: Plaintext over the SDIO interface • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: FIPS 186-4 Key Generation on Perform Key Agreement Process service request
Remote Party Diffie-Hellman Ephemeral Public Key	Used to establish a shared secret over an insecure channel. • Entry: Plaintext over SDIO interface • Output: N/A • Storage: Plaintext in volatile memory • Zeroization: Delete Key Variable, Power cycle • Generation: N/A

4 Roles, Authentication and Services

4.1 Assumption of Roles

The Module supports a User, KVL and Cryptographic Officer (CO) role. Authentication data is initialized to a default value in manufacturing. After authenticating, the Crypto-Officer and User passwords may be changed at any time. The Module enforces the separation of roles using login credentials. Re-authentication is enforced when changing roles.

Table 11 lists all operator roles supported by the Module. The Module does not support a maintenance role or bypass capability. The Module does not support concurrent operators.

Table 12 – Roles Description

Role ID	Role Description	Authentication Type	Authentication Data
CO	Cryptographic Officer Role over SDIO interface.	Identity-based	14-32 character ASCII password
User	User Role over SDIO interface.	Identity-based	14-32 character ASCII password
KVL	When a User or CO is connected to SDIO through a Motorola KVL device.	Role-based	256-bit AES key (KVL-BKK)

4.2 Authentication Methods

Password Authentication

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 takes approximately 4.19ms to authenticate over the SDIO interface. The worst-case probability of a successful random attempt within a one-minute period is 14320/95^14 which is less than 1 in 1,000,000.

After a configurable number of consecutive failed attempts, the KPK, TEKs and KEKs are zeroized, a new KPK is generated, and the password is reset to the factory default. Note that this makes it very important that physical access to the Module is strictly controlled. The module is not usable until the factory default password is changed.

KVL-BKK Authentication:

Communications between the Module and a KVL device are encrypted with the 256-bit KVL-BKK. A KVL device is authenticated by having possession of the key needed to decrypt communications. The probability of a successful random attempt is 1 in 2^{256} which is less than 1 in 1,000,000.

It takes approximately 80.5 milliseconds for each encrypted data packet to be sent between the Module and KVL. Therefore; the maximum number of authentication attempts that can be performed as a KVL Role with the KVL-BKK in one minute is 745 and the probability of a successful random attempt during a one-minute period is 745 in 2^{256} or 1 in 1.55425e+74.

Table 13 – Authentication Description

Authentication Method	Probability	Justification
Password	$1/95^{14}$	$15/95^{14}$
KVL-BKK	$1/2^{256}$	$745/2^{256}$

4.3 Services

All services implemented by the Module are listed in the tables below. Note that all services listed in Tables 13 and 14 below are available in both the FIPS Approved and non-Approved mode. The only distinguishing factor between Approved and non-Approved services is whether non-Approved algorithms/ key establishment schemes are available.

Table 14 – Authenticated Services

Service	Description	CO	User	KVL
Program Update	Update the Module firmware via the SDIO interface. All keys (stored in volatile and non-volatile memory) and CSPs may be zeroized during a Program Update.	X	X	X
Validate Crypto-Officer password	Validate the current Crypto-Officer password used to identify and authenticate the Crypto-Officer role via the SDIO interface. Successful authentication will allow access to services allowed for the Crypto Officer.	X		
Change Crypto-Officer password	Modify the current password used to identify and authenticate the Crypto-Officer Role via SDIO interface.	X		
Extract Action Log	Exports a history of actions over the SDIO interface.	X	X	
Logout Crypto-Officer Role	Logs out the Crypto-Officer.	X		
Configure Module via SDIO interface	Perform configuration of the Module (e.g. time configuration, enable/disable clear key import, enable/disable red keyloading, etc.) via the SDIO interface.	X		

Service	Description	CO	User	KVL
Validate User Password	Validate the current User password used to identify and authenticate the User role via the SDIO interface. Successful authentication will allow access to crypto services allowed for the User.		X	
Change User Password	Modify the current password used to identify and authenticate the User Role via the SDIO interface.		X	
Algorithm List Query	Provides a list of algorithms over the SDIO interface.		X	
Logout User Role	Logs out the User.		X	
Export Key Variable	Transfer encrypted key variables (e.g. KEKs, TEKs) out of the Module over the SDIO interface. Transfer clear key variables out of the Module over SDIO interface is supported when the Module is running in non-FIPS mode.		X	
Import Key Variable	Receive encrypted key variables (e.g. KEKs, and TEKs) over the SDIO and KVL interfaces. Receive clear key variables over SDIO interface is supported when the Module is running in non-FIPS mode.		X	X
Generate Key Variable	Auto-generate Public and Private Generated Signature Keys, SRTP/SRTCP Master Key, SRTP/ SRTCP Master Salt, TLS Master Secret Key and the KPK within the Module.		X	
Delete Key Variable	Delete KEKs, TEKs, ECDH Public and Private Keys, ECDH Public and Private Generated Signature Keys, and ECDH Shared Secret.		X	
Encrypt	Encrypt plaintext data to be transferred over the SDIO interface.		X	
Decrypt	Decrypt ciphertext data received over the SDIO interface.		X	
Generate Signature	Generate a Signature and output result over SDIO interface.		X	
Verify Signature	Verify a Signature and output result over SDIO interface.		X	
Generate Hash	Generate a hash and output result over SDIO interface.		X	
Generate MAC	Generate a Message Authentication Code of a block of data to provide data integrity using a shared symmetric key.		X	
Perform Key Agreement Process	Perform a key agreement process over SDIO interface		X	
Generate Random Number	Generate random data using NDRNG and output result over SDIO interface.		X	
Key Query	Retrieve the metadata for a given key present in the Module.		X	

Service	Description	CO	User	KVL
OTAR	Modify and query the TEKs and KEKs in the Module via APCO OTAR Key Management Messages.		X	
Zeroize Keys via KVL interface	Zeroize KEKs and TEKs when connected to the KVL.			X
Configure Module via KVL interface	Perform configuration of the Module (e.g. OTAR configuration) when connected to the KVL.			X

Table 15 – Unauthenticated Services

Service	Description
Perform Self-Tests	Performs module self-tests comprised of cryptographic algorithms test and firmware test. Initiated by a transition from power off state to power on state.
Version Query	Provides module firmware version number and FIPS status over the SDIO interface.

Table 15 defines the relationship between access to Security Parameters and the different module services. The modes of access shown in the table are defined as:

- C = Check CSP: Check status of the CSP (i.e. existence, size, format, etc.).
- D = Decrypt: Decrypts entered key using the BKK during CSP entry over the SDIO interface or using the KVL-BKK during CSP entry over the KVL interface. In the case of the Program Update service, decryption will occur using the IDK.
- E = Encrypt: Encrypts key prior to output over the SDIO interface using a KEK.
- G = Generate CSP: Generates key.
- S = Store CSP: Stores CSP in volatile or non-volatile memory.
- U = Use CSP: Uses key internally for encryption/decryption services.
- Z = Zeroize: The service zeroizes the CSP.
- - = No access: the service does not access the CSP.

Table 16 – Security Parameters Access by Service

Service	CSPs and Public Keys																			
	PEK	TEKs	KEKs	KPK	KVL-BKK	BKK	IDK	UKPPK	Crypto-Officer Password	User Password	ECDSA Private Generated Signature Key	ECDSA Public Programmed Signature Key	ECDSA Public Generated Signature Key	ECDH Private Key	ECDH Shared Secret	ECDH Public Key	Remote Party DHE Public Key	SRTP/SRTCP Master Key	SRTP/SRTCP Master Salt	TLS KDF Secret	KDF Derived Key
Program Update	D,Z,S	Z	Z	Z	D,Z,S	D,Z,S	U,Z,S	D,Z	Z	Z	Z, U	Z, U	Z, U	Z,U	Z	Z	Z	Z	Z	Z	Z
Validate Crypto-Officer password	U	-	-	D	-	-	-	U	D,U,Z	-	-	-	-	-	-	-	-	-	-	-	-
Change Crypto-Officer password	U	-	-	D,S,E,G	-	-	-	U	D,U,Z,S	-	-	-	-	-	-	-	-	-	-	-	-
Logout Crypto-Officer Role	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Extract Action Log	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Configure Module via SDIO interface	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Validate User Password	U	-	-	D	-	-	-	U	-	D,U,Z	-	-	-	-	-	-	-	-	-	-	-
Change User Password	U	-	-	-	-	-	-	U	-	D,U,Z,S	-	-	-	-	-	-	-	-	-	-	-
Logout User Role	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Algorithm List Query	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Export Key Variable	-	D,E,U	D,E,U	U	-	-	-	-	-	-		-	-	-	D,E,U	-	-	D,E,U	D,E,U	D,E,U	D,E,U
Import Key Variable	-	D,E,S,U	D,E,S,U	U	U	U	-	-	-	-	-	-	-	U	-	-	-	D,E,S,U	D,E,S,U	D,E,S,U	D,E,S,U
Generate Key Variable	-			U	-	-	-	-	-	-	E,G,S		E,G,S	-	-	-	-	E,G,S	E,G,S	E,G,S	E,G,S
Delete Key Variable	-	Z	Z	-	-	-	-	-	-	-	Z	-	-	Z	Z	-	-	Z	Z	Z	Z
Encrypt	-	C,U	C,U	C,U	C,U	C,U	-	-	-	-	-	-	-	-	-	-	-	C,U	C,U	C,U	C,U
Decrypt	-	C,U	C,U	C,U	C,U	C,U	-	-	-	-	-	-	-	U	-	-	-	C,U	C,U	C,U	C,U
Generate Signature	-	-	-	U	-	-	-	-	-	-	D,U	-	-	U	U	-	-	-	-	-	-
Verify Signature	-	-	-	U	-	-	-	-	-	-	-	U	U	U	U	-	-	-	-	-	-
Generate Hash	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generate MAC	-	D,U,C	-	U	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	U
Perform Key Agreement Process	-	E,S	E,S	-	-	-	-	-	-	-	E,G,S	-	-	E,G,S	U	G,U	U	-	-	-	-
Generate Random Number	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Key Query	-	D	D	U	-	-	-	-	-	-	-	-	-	-	-	-	-	U	U	U	U

Service	CSPs and Public Keys																				
	PEK	TEKs	KEKs	KPK	KVL-BKK	BKK	IDK	UKPPK	Crypto-Officer Password	User Password	ECDSA Private Generated Signature Key	ECDSA Public Programmed Signature Key	ECDSA Public Generated Signature Key	ECDH Private Key	ECDH Shared Secret	ECDH Public Key	Remote Party DHE Public Key	SRTP/SRTCP Master Key	SRTP/SRTCP Master Salt	TLS KDF Secret	KDF Derived Key
OTAR	-	D,E,S, U	D,E,S, U	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Zeroize Keys via KVL interface	-	Z	Z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Perform Self-Tests	-	-	-	-	-	-	-	-	-	-	-	U	-	-	-	-	-	-	-	-	-
Version Query	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

5 Self-tests

The Module performs self-tests to ensure the proper operation of the Module. Per FIPS 140-2 these are categorized as either power-up self-tests or conditional self-tests. Power-up self–tests are available on demand by power cycling the Module.

All algorithm Known Answer Tests (KATs) must be completed successfully prior to any other use of cryptographic functionality by the Module. The Module toggles the Self-test Indicator interface within 2 seconds of power-up to indicate the Firmware Integrity Test, Firmware Load Test, Cryptographic Algorithm Tests, and Critical Functions Test have completed successfully. The Module enters the Critical Error state and does not toggle the Self-test Indicator interface if the Firmware Integrity Test, Firmware Load Test, Cryptographic Algorithm Tests, or Critical Functions Test fails. The Critical Error state may be exited by powering the Module off then on.

The Module performs the following algorithm KATs on power-up.

- Firmware Integrity: A digital signature is generated over the base firmware and all Drop-in algorithms code when it is built using SHA-384 (Cert. #4132) and ECDSA P-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.
- AES-128 encrypt and decrypt KATs for CTR, ECB, OFB, and CBC modes.
- AES-256 encrypt and decrypt KATs for CTR, ECB, OFB, CBC, GCM and CFB modes.
- ECDSA P-384 key generation KAT
- ECDSA P-384 signature generation and verification KATs.
- Diffie-Hellman primitive “Z” computation KAT per IG 9.6.
- SHA-256 and -384 KAT (Cert. #4132).
- SHA-256 and -384 KAT (Cert. #4133).
- HMAC-384 KAT (Cert. #3386).
- HMAC-384 KAT (Cert. #3387).

The Module performs the following critical functions tests as indicated.

- The Module performs a read/write test of the internal RAM at each power up.
- Random Number Generator entropy test. This test runs two RNG statistical tests: a FIPS monobit test, and a FIPS “runs” test as defined in SP 800-22r1a.

The Module performs the following conditional self-tests as indicated.

- ECDSA Pairwise consistency test on ECDSA key pair generation: The ECDSA Public and Private Generated Signature Key pair is tested by the calculation and verification of a digital signature. If the digital signature cannot be verified, the test fails.
- Continuous Random Number Generator test: The continuous random number generator test is performed on NDRNG supported by the Module. 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 NDRNG 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. This testing is done for each 4 byte RNG data block, generated by the NDRNG. The Module enters the Critical Error State if this test fails.
- Firmware load test: a digital signature is generated over the code when it is built using SHA-384 (Cert. #4132) and ECDSA P-384. Upon download into the module, the digital signature is verified. If the digital signature matches, then the test passes, otherwise it fails.

6 Physical Security Policy

The Module is a production grade, single chip cryptographic module as defined by FIPS 140-2 and is designed to meet Level 3 Physical Security.

The Module 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 cryptographic services will be available.

No maintenance access interface is available.

7 Operational Environment

The Module has a non-modifiable operational environment under the FIPS 140-2 definitions. The Module includes Program Update service to support necessary updates. Firmware versions validated through the FIPS 140-2 CMVP will be explicitly identified on a validation certificate. If firmware that is not identified in this Security Policy is loaded into the Module, the Module will be in a non-Approved mode.

8 Mitigation of Other Attacks Policy

The Module is not designed to mitigate any specific attacks outside of those required by FIPS 140-2.

9 Security Rules and Guidance

This section documents the security rules for the secure operation of the Module to implement the security requirements of FIPS 140-2.

9.1 Invariant Rules

1. An operator does not have access to any cryptographic services prior to assuming an authorized role.
2. Power up self-tests do not require any operator action.
3. Data output is inhibited during key generation, self-tests, zeroization, and while in critical error states.
4. The Module does not perform any cryptographic functions while in an error state.
5. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.
6. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
7. The Module does not support manual key entry.
8. The Module does not enter or output plaintext CSPs in the Approved mode.
9. The Module implements all firmware using a high-level language, except the limited use of low-level languages to enhance performance.
10. The Module conforms to FCC 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B requirements.

9.2 Procedural Enforcement

1. An operator shall ensure that the security strength of the TLS KDF Secret is at least as strong as the length of the resulting KDF Derived Key.
2. An operator shall ensure KDF Derived Keys used in the FIPS Approved mode have at least 112 bits of security strength
3. An operator shall ensure KDF Derived Keys used for key transport have at least the security strength of the key(s) being transported.
4. An operator shall ensure KDF Derived keys are only used within the context of the TLS or SRTP/ SRTCP protocols, dependent on which protocol KDF was used to derive the key.
5. An operator shall not output a KDF Derived Key in plaintext.
6. If the module fails the FW integrity test, an operator shall ship the module to a Motorola service center for recovery.
7. The module does not restrict the usage of keys to a single purpose (e.g., an operator is capable of choosing to use the SRTP/ SRTCP Master Key in place of a KEK in the OTAR service). While in the Approved mode, an operator shall only use keys for their designated purpose.

8. The module is capable of using the BKK within the context of the Export Key Variable service to output keys encrypted with non-Approved key wrapping (AES-OFB encryption without an Approved MAC authentication) as follows:

   TEKs, KEKs, SRTP/ SRTCP Master Key, SRTP/ SRTCP Master Salt, KDF Derived Key

   An operator shall ensure that non-Approved key wrapping is not used in the Approved mode of operation.

10 AES-256 GCM IV Generation Protocol

The Module generates GCM IVs deterministically as specified in SP800-38D Section 8.2.1 using the following protocols:

- TLS: The Module is compliant with RFC 5288, which implies compliance with TLS v1.2 and SP800-52 Rev1, Section 3.3.1 as required per IG A.5. The fixed field consists of a 16-bit salt that is generated internally to the Module and the invocation field consists of a 64-bit nonce_explicit passed into the Module as an input parameter.
- SRTP: The Module is compliant with RFC 7714, Section 8.1 IV construction. The fixed field consists of a 32-bit Synchronization Source identifier and 16-bits of zeroes, and the invocation field consists of a 16-bit Sequence Number and 32-bit Rollover Counter. Both the fixed field and invocation field are passed into the Module as input parameters and XORed with a 96-bit random salt imported or generated internally. Note that the XOR operation does not have an impact on SP 800-38D requirements because the salt is not regenerated until a key is re-established and therefore acts as a constant within an individual key's lifecycle.
- SRTCP: The Module is compliant with RFC 7714, Section 9.1 IV construction. The fixed field consists of a 32-bit Synchronization Source and 17 bits of zeroes, and the invocation field consists of a 31-bit SRTCP Index. Both the fixed field and invocation field are passed into the Module as input parameters and XORed with a 96-bit random salt imported or generated internally. Note that the XOR operation does not have an impact on SP 800-38D requirements because the salt is not regenerated until a key is re-established and therefore acts as a constant within an individual key's lifecycle.

If the Module's power is lost and restored for any of the protocols listed above, a new GCM key will be established. The invocation field is incremented externally and input to the Module; if the new invocation field is not greater than the last value then the Module will transition to an error state. Following an overflow of the invocation field, the Module will transition to an error state.

11 References and Definitions

The following standards are referred to in this Security Policy.

Table 17 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, May 25, 2001
[IG]	*Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*
[108]	*NIST Special Publication 800-108, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009*
[131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, *March 2019*
[133]	*NIST Special Publication 800-133, Recommendation for Cryptographic Key Generation, December 2012*
[135]	*National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011.*
[186]	*National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July 2013.*
[197]	*National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001*
[198]	*National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008*
[180]	*National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015*
[22r1a]	*National Institute of Standards and Technology, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, April 2010*
[38A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001*
[38B]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Special Publication 800-38B, May 2005*
[38D]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Special Publication 800-38D, November 2007*

Abbreviation	Full Specification Name
[38F]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, Special Publication 800-38F, December 2012*
[56A]	*NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, April 2018*
[OTAR]	*Project 25 - Digital Radio Over-The-Air-Rekeying (OTAR) Messages and Procedures, September 2014*
[RFC2246]	*The TLS Protocol, August 2008*
[RFC3711]	*The Secure Real-time Transport Protocol (SRTP), March 2004*
[RFC5286]	*AES Galois Counter Mode (GCM) Cipher Suites for TLS, August 2008*
[RFC5246]	*The Transport Layer Security (TLS) Protocol, August 2008*
[RFC7714]	*AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP), December 2015*

Table 18 – Acronyms and Definitions

Acronym	Definition
AES	Advanced Encryption Standard
AME	Assured Mobile Environment
BKK	Black Keyloading Key
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CSP	Critical Security Parameter
ECB	Electronic Code Book
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FW	Firmware
GCM	Galois/Counter Mode
IDK	Image Decryption Key
IV	Initialization Vector
KDF	Key Derivation Function
KLK	Key Loss Key
KPK	Key Protection Key
KEK	Key Encryption Key

Acronym	Definition
KVL	Key Variable Loader
KVL-BKK	KVL - Black Keyloading Key
LTE	Long Term Evolution
OTAR	Over The Air Rekeying
PEK	Password Encryption Key
PGSK	Private Generated Signature Key
NDRNG	Non-Deterministic Random Number Generator
SRTP	Secure Real-time Transport Protocol
SRTCP	Secure Real-time Transport Control Protocol
TEK	Traffic Encryption Key
TLS	Transport Layer Security
UKPPK	Universal Key Protection Protection Key