Crypto4A Technologies Inc.

QASM Cryptographic Module

# FIPS 140-3 Non-Proprietary Security Policy

# Table of Contents

# List of Tables



# List of Figures

# 1 General

## 1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 1.1 of the QASM Cryptographic Module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 3 module.

## 1.2 Security Levels

| Section | Title | Security Level |
|---|---|---|
| 1 | General | 3 |
| 2 | Cryptographic module specification | 3 |
| 3 | Cryptographic module interfaces | 3 |
| 4 | Roles, services, and authentication | 3 |
| 5 | Software/Firmware security | 3 |
| 6 | Operational environment | N/A |
| 7 | Physical security | 3 |
| 8 | Non-invasive security | N/A |
| 9 | Sensitive security parameter management | 3 |
| 10 | Self-tests | 3 |
| 11 | Life-cycle assurance | 3 |
| 12 | Mitigation of other attacks | N/A |
| | Overall Level | 3 |

Table 1: Security Levels

## 1.3 Additional Information

The QASM Cryptographic Module is a high-assurance, tamper-resistant Hardware Security Module (HSM) which secures sensitive data and critical applications by storing, protecting and managing cryptographic keys. It provides end users with industry-leading security and performance. The QASM can be embedded into either the QxHSM or QxEDGE families of security appliances for FIPS 140-3 validated key security.

The QxHSM is a blade-based appliance form factor that can be inserted into a variety of chassis modules:

- Desktop-oriented QxBMC-1
- 1U 19-inch rack-mountable QxBMC-3, which can contain up to 3 QxHSM devices
- 4U 19-inch rack-mountable QxBMC-12 high density chassis capable of housing up to 12 QxHSM devices.

The QxHSM comprises a Single Board Computer (SBC) and a QASM HSM. The QxHSM SBC is based on an Intel x86 Xeon processor and is typically configured with 16 GB of RAM and 256 GB of SSD storage. The SBC implements a hardened Linux-based Operating System (OS) that has a direct connection to the QASM, as well as dual redundant RJ45 1GB network interfaces.

The QxEDGE is a 1U, 19-inch rack mountable appliance form factor that provides a great deal more computing power than the more modular QxHSM platform. In its base configuration the QxEDGE comprises 4 SBCs and a single central QASM HSM module. The SBCs are configured as in the QxHSM, with each given their own dedicated and separate connection to the QASM module. This allows local software applications running on these independent SBCs to interact with the QASM as four independent computing systems, enabling more complex compute platform configurations. Hence, the QxEDGE is intended for more elaborate use cases beyond the simple network-connected HSM use case that the QxHSM is targeting.

# 2 Cryptographic Module Specification

## 2.1 Description

**Purpose and Use:**

The QASM Cryptographic Module (hereafter referred to as “the module”) is a Hardware Security Module (HSM) that provides management of sensitive information and quantum-resistant cryptographic services, all within a secure anti-tamper housing.

**Module Type**: Hardware

**Module Embodiment**: Multi-Chip Standalone

**Cryptographic Boundary:**

The cryptographic boundary of the module is defined by the aluminum housing around the device.

**Tested Operational Environment’s Physical Perimeter (TOEPP):**

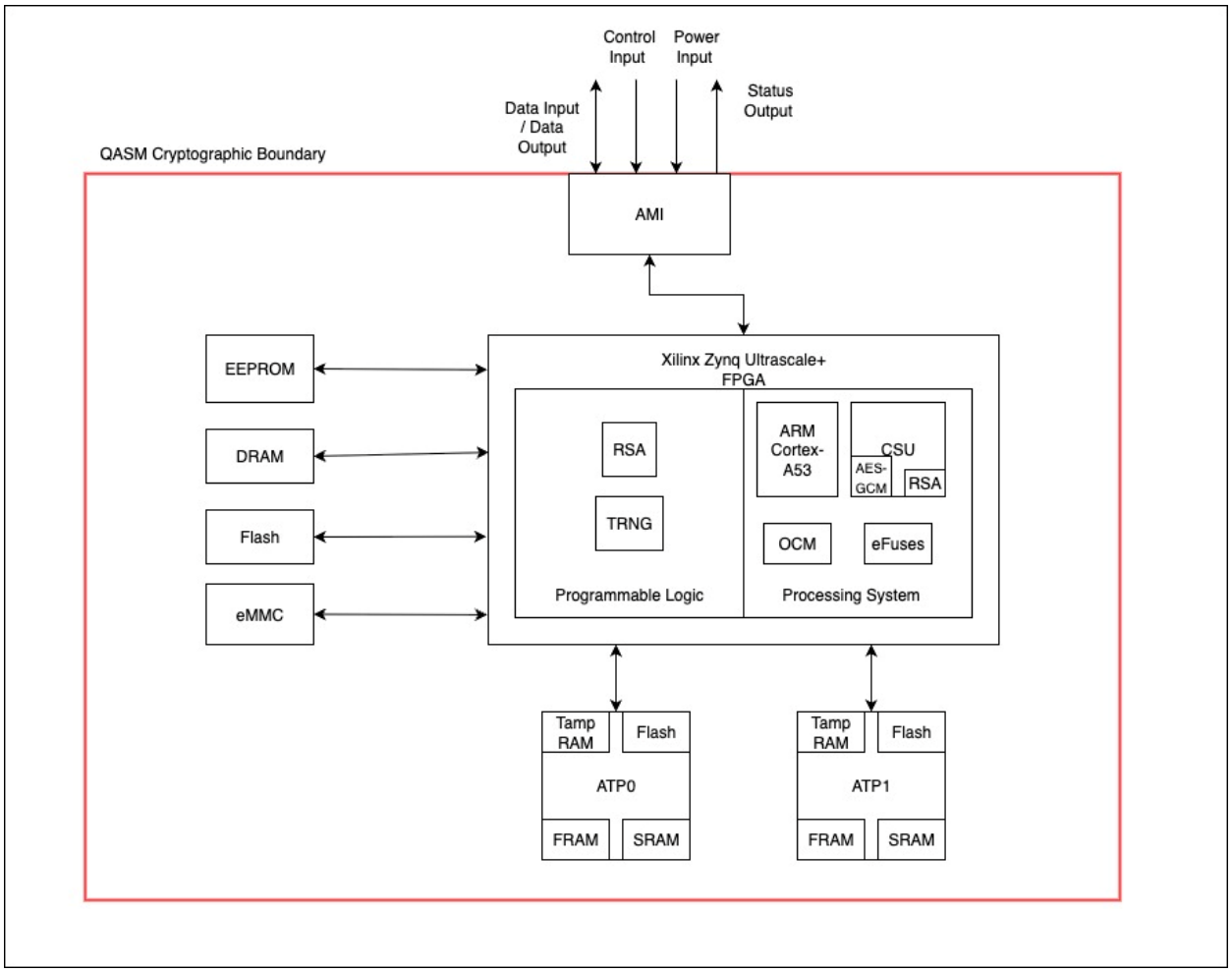
The hard enclosure containing the QASM forms the TOEPP.

Figures 1 and 2 shows pictures of the physical module. Figure 3 shows the block diagram with all major components and all interfaces of the module.

Figure 1: QASM top view

Figure 2: QASM bottom view


Figure 3: Block Diagram



## 2.2 Tested and Vendor Affirmed Module Version and Identification

To retrieve the module version, the operator shall use the Show Version service by running the "skm hsm info" command and verify that the output matches the information in Table 2. Specifically, the output shall indicate Module Version: QASM 1.1, Firmware Version: 5.0.1.158, and Operational Mode: FIPS. If any version value differs, or if the Operational Mode is reported as STANDARD instead of FIPS, the module is not operating in the validated configuration and it is considered as a non-validated module.

**Tested Module Identification – Hardware:**

| Model and/or Part Number | Hardware Version | Firmware Version | Processors | Features |
|---|---|---|---|---|
| FIPS | QASM 1.1 and Operational Mode: FIPS | 5.0.1.158 | Xilinx Zynq Ultrascale+ ZU9EG | N/A |

Table 2: Tested Module Identification – Hardware

## 2.3 Excluded Components

N/A, the module does not have any excluded components.

## 2.4 Modes of Operation

**Modes List and Description:**

| Mode Name | Description | Type | Status Indicator |
|---|---|---|---|
| Approved Mode | The module is only capable of providing approved services, and conforms to FIPS standards. | Approved | Successful completion of a service is an implicit indicator for the use of an approved service. |

Table 3: Modes List and Description

When the module is powered on, initialization automatically begins. Once all self-tests and integrity tests are complete, the module transitions directly into the approved mode until a reboot occurs.

## 2.5 Algorithms

**Approved Algorithms:**

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| AES-CBC | A5631 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |
| AES-ECB | A5631 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| AES-GCM | A5631 | Direction - Decrypt, Encrypt<br>IV Generation - Internal<br>IV Generation Mode - 8.2.2<br>Key Length - 128, 192, 256<br>Tag Length - 104, 112, 120, 128, 32, 64, 96<br>IV Length - IV Length: 96<br>Payload Length - Payload Length: 8-65536 Increment 8<br>AAD Length - AAD Length: 0-65536 Increment 8 | SP 800-38D |
| AES-GCM | A6119 | Direction - Decrypt, Encrypt<br>IV Generation - Internal<br>IV Generation Mode - 8.2.2<br>Key Length - 256<br>Tag Length - 128<br>IV Length - IV Length: 96<br>Payload Length - Payload Length: 32-65536 Increment 32<br>AAD Length - AAD Length: 0 | SP 800-38D |
| AES-KW | A5631 | Direction - Decrypt, Encrypt<br>Cipher - Cipher<br>Key Length - 128, 192, 256<br>Payload Length - Payload Length: 128-4096 Increment 128 | SP 800-38F |
| AES-KWP | A5631 | Direction - Decrypt, Encrypt<br>Cipher - Cipher<br>Key Length - 128, 192, 256<br>Payload Length - Payload Length: 8-4096 Increment 8 | SP 800-38F |
| ECDSA KeyGen (FIPS186-5) | A5631 | Curve - P-224, P-256, P-384, P-521<br>Secret Generation Mode - extra bits, testing candidates | FIPS 186-5 |
| ECDSA KeyVer (FIPS186-5) | A5631 | Curve - P-224, P-256, P-384, P-521 | FIPS 186-5 |
| ECDSA SigGen (FIPS186-5) | A5631 | Curve - P-224, P-256, P-384, P-521<br>Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512<br>Component - No, Yes | FIPS 186-5 |
| ECDSA SigVer (FIPS186-5) | A5631 | Component - No<br>Curve - P-224, P-256, P-384, P-521<br>Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 | FIPS 186-5 |
| EDDSA KeyGen | A5631 | Curve - ED-25519, ED-448 | FIPS 186-5 |
| EDDSA KeyVer | A5631 | Curve - ED-25519, ED-448 | FIPS 186-5 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| EDDSA SigGen | A5631 | Curve - ED-25519, ED-448<br>PreHash - No<br>Pure - Yes<br>Context Length - Context Length: 0-255 Increment 1 | FIPS 186-5 |
| EDDSA SigVer | A5631 | Curve - ED-25519, ED-448<br>PreHash - No<br>Pure - Yes | FIPS 186-5 |
| Hash DRBG | A5631 | Prediction Resistance - No, Yes<br>Supports Reseed - Yes<br>Mode - SHA2-512<br>Entropy Input - Entropy Input: 256<br>Nonce - Nonce: 128<br>Personalization String Length - Personalization String Length: 256<br>Additional Input - Additional Input: 256<br>Returned Bits - 512 | SP 800-90A Rev. 1 |
| HMAC-SHA-1 | A5631 | MAC - MAC: 32-160 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-224 | A5631 | MAC - MAC: 32-224 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-256 | A5631 | MAC - MAC: 32-256 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-384 | A5631 | MAC - MAC: 32-384 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-512 | A5631 | MAC - MAC: 32-512 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-224 | A5631 | MAC - MAC: 32-224 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-256 | A5631 | MAC - MAC: 32-256 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-512 | A5631 | MAC - MAC: 32-512 Increment 8<br>Key Length - Key Length: 8-8192 Increment 8 | FIPS 198-1 |
| KAS-ECC Sp800-56Ar3 | A5631 | Domain Parameter Generation Methods - P-224, P-256, P-384, P-521<br>Function - Full Validation, Key Pair Generation<br>iutId - 1234567890abcdef<br>Scheme -<br>onePassDh -<br>KAS Role - Responder<br>KDF Methods -<br>oneStepKdf -<br>Auxiliary Function Methods -<br>Auxiliary Function Name - SHA2-224<br>Fixed Info Pattern - uPartyInfo\|\|vPartyInfo<br>Fixed Info Encoding - Concatenation<br>Key Length - 256 | SP 800-56A Rev. 3 |
| KDA OneStep SP800-56Cr2 | A5631 | Auxiliary Function Methods -<br>Auxiliary Function Name - SHA2-224 | SP 800-56C Rev. 2 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| | | Fixed Info Pattern - uPartyInfo\|\|vPartyInfo<br>Fixed Info Encoding - concatenation<br>Derived Key Length - 256<br>Shared Secret Length - Shared Secret Length: 256-1024 Increment 256 | |
| KDF ANS 9.63 (CVL) | A5631 | Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512<br>Field Size - 224<br>Shared Info Length - Shared Info Length: 0-1024 Increment 8<br>Key Data Length - Key Data Length: 128-4096 Increment 8 | SP 800-135 Rev. 1 |
| KDF SP800-108 | A5631 | KDF Mode - Counter<br>MAC Mode - HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512<br>Supported Lengths - Supported Lengths: 8-4096 Increment 8<br>Fixed Data Order - Before Fixed Data<br>Counter Length - 32<br>Supports Empty IV - No<br>Requires Empty IV - No<br>Custom Key In Length - 0 | SP 800-108 Rev. 1 |
| KTS-IFC | A5631 | Function - keyPairGen<br>IUT ID - 0123456789abcdef<br>Modulo - 2048, 3072, 4096, 8192<br>Key Generation Methods - rsakpg1-basic<br>Fixed Public Exponent - 010001<br>Scheme -<br>KTS-OAEP-basic -<br>KAS Role - initiator, responder<br>Key Transport Method -<br>Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512<br>Supports Null Associated Data - Yes<br>Key Length - 256 | SP 800-56B Rev. 2 |
| KTS-IFC | A6409 | Function - keyPairGen<br>IUT ID - 0123456789abcdef<br>Modulo - 2048, 3072, 4096<br>Key Generation Methods - rsakpg1-basic<br>Fixed Public Exponent - 010001<br>Scheme -<br>KTS-OAEP-basic -<br>KAS Role - initiator, responder<br>Key Transport Method - | SP 800-56B Rev. 2 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| | | Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512<br>Supports Null Associated Data - Yes<br>Key Length - 256 | |
| KTS-IFC | A6410 | Function - keyPairGen<br>IUT ID - 0123456789abcdef<br>Modulo - 2048, 3072, 4096<br>Key Generation Methods - rsakpg1-basic<br>Fixed Public Exponent - 010001<br>Scheme -<br>KTS-OAEP-basic -<br>KAS Role - initiator, responder<br>Key Transport Method -<br>Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512<br>Supports Null Associated Data - Yes<br>Key Length - 256 | SP 800-56B Rev. 2 |
| LMS KeyGen | A5631 | Specific Capabilities -<br>LMS Modes - LMS_SHA256_M24_H5<br>LMOTS Modes - LMOTS_SHA256_N24_W2 | SP 800-208 |
| LMS SigGen | A5631 | Specific Capabilities -<br>LMS Modes - LMS_SHA256_M24_H5<br>LMOTS Modes - LMOTS_SHA256_N24_W2 | SP 800-208 |
| LMS SigVer | A5631 | Specific Capabilities -<br>LMS Modes - LMS_SHA256_M24_H5<br>LMOTS Modes - LMOTS_SHA256_N24_W1 | SP 800-208 |
| ML-DSA KeyGen | A5631 | Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87 | FIPS 204 |
| ML-DSA SigGen | A5631 | Signature Interfaces - external, internal<br>Deterministic - No, Yes<br>External Mu - No, Yes<br>Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Message Length - Message Length: 8-65536 Increment 8<br>Pre Hash - preHash, pure<br>Hash Algorithms - SHA2-512<br>Context Length - Context Length: 0-2040 Increment 8 | FIPS 204 |
| ML-DSA SigVer | A5631 | Signature Interfaces - external, internal<br>External Mu - No, Yes<br>Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Message Length - Message Length: 8-65536 Increment 8<br>Pre Hash - preHash, pure | FIPS 204 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| | | Hash Algorithms - SHA2-512<br>Context Length - Context Length: 0-2040 Increment 8 | |
| ML-KEM EncapDecap | A5631 | Parameter Sets - ML-KEM-1024, ML-KEM-512, ML-KEM-768<br>Functions - Decapsulation, Encapsulation | FIPS 203 |
| ML-KEM KeyGen | A5631 | Parameter Sets - ML-KEM-1024, ML-KEM-512, ML-KEM-768 | FIPS 203 |
| RSA KeyGen (FIPS186-5) | A5631 | Key Generation Mode - probable<br>Modulo - 2048, 3072, 4096, 8192<br>p mod 8 - 0<br>Primality Tests - 2powSecStr<br>q mod 8 - 0<br>Fixed Public Exponent - 010001<br>Info Generated By Server - Yes<br>Private Key Format - standard<br>Public Exponent Mode - fixed | FIPS 186-5 |
| RSA SigGen (FIPS186-5) | A5631 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1 | FIPS 186-5 |
| RSA SigGen (FIPS186-5) | A6409 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1 | FIPS 186-5 |
| RSA SigGen (FIPS186-5) | A6410 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1 | FIPS 186-5 |
| RSA Signature Primitive (CVL) | A5631 | Modulo - 2048, 3072, 4096<br>Private Key Format - standard<br>Public Exponent Mode - fixed<br>Fixed Public Exponent - 010001 | FIPS 186-4 |
| RSA Signature Primitive (CVL) | A6409 | Modulo - 2048, 3072, 4096<br>Private Key Format - standard<br>Public Exponent Mode - fixed<br>Fixed Public Exponent - 010001 | FIPS 186-4 |
| RSA Signature Primitive (CVL) | A6410 | Modulo - 2048, 3072, 4096<br>Private Key Format - standard<br>Public Exponent Mode - fixed<br>Fixed Public Exponent - 010001 | FIPS 186-4 |
| RSA SigVer (FIPS186-5) | A5631 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1 | FIPS 186-5 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| | | Fixed Public Exponent - 010001<br>Public Exponent Mode - fixed | |
| RSA SigVer (FIPS186-5) | A6409 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1<br>Fixed Public Exponent - 010001<br>Public Exponent Mode - fixed | FIPS 186-5 |
| RSA SigVer (FIPS186-5) | A6410 | Hash Pair -<br>Hash Algorithm - SHA2-224<br>Modulo - 2048, 3072, 4096<br>Signature Type - pkcs1v1.5, pss<br>Mask Function - mgf1<br>Fixed Public Exponent - 010001<br>Public Exponent Mode - fixed | FIPS 186-5 |
| SHA-1 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-224 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-256 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-384 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-512 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-512/224 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA2-512/256 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 180-4 |
| SHA3-224 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 202 |
| SHA3-256 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 202 |
| SHA3-384 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 202 |
| SHA3-512 | A5631 | Message Length - Message Length: 0-65536 Increment 8 | FIPS 202 |
| SHAKE-128 | A5631 | Supports Bit-Oriented Messages - No<br>Supports Empty Message - Yes<br>Supports Bit-Oriented Output - No<br>Output Length - Output Length: 16-16384 Increment 8 | FIPS 202 |
| SHAKE-256 | A5631 | Supports Bit-Oriented Messages - No<br>Supports Empty Message - Yes<br>Supports Bit-Oriented Output - No<br>Output Length - Output Length: 16-16384 Increment 8 | FIPS 202 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| SLH-DSA KeyGen | A5631 | Parameter Sets - SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | FIPS 205 |
| SLH-DSA SigGen | A5631 | Deterministic - No, Yes<br>Signature Interfaces - external, internal<br>Parameter Sets - SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Message Length - Message Length: 8-65536 Increment 8<br>Pre Hash - preHash, pure<br>Hash Algorithms - SHA2-256, SHA2-512, SHAKE-128, SHAKE-256<br>Context Length - Context Length: 0-2040 Increment 8 | FIPS 205 |
| SLH-DSA SigVer | A5631 | Signature Interfaces - external, internal<br>Parameter Sets - SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Message Length - Message Length: 8-65536 Increment 8<br>Pre Hash - preHash, pure<br>Hash Algorithms - SHA2-256, SHA2-512, SHAKE-128, SHAKE-256<br>Context Length - Context Length: 0-2040 Increment 8 | FIPS 205 |

Table 4: Approved Algorithms

**Vendor-Affirmed Algorithms:**

| Name | Properties | Implementation | Reference |
|---|---|---|---|
| HSS SigGen | Key Type:Asymmetric | N/A | The LMS operations used by the HSS implementation were CAVP tested in accordance with IG C.O with Cert A5631 |
| HSS SigVer | Key Type:Asymmetric | N/A | The LMS operations used by the HSS implementation were CAVP tested in accordance with IG C.O with Cert A5631 |
| HSS KeyGen | Key Type:Asymmetric | N/A | The LMS operations used by the HSS implementation were CAVP tested in |

| Name | Properties | Implementation | Reference |
|---|---|---|---|
| | | | accordance with IG C.O with Cert A5631 |
| CKG | Key Type:Symmetric and Asymmetric | N/A | following SP800-133rev2 section 4 example 1, B = U |
| CKG2 | Key Type:Symmetric | N/A | following SP800-133rev2 section 6.3 method 2 |

Table 5: Vendor-Affirmed Algorithms

**Non-Approved, Allowed Algorithms:**

N/A for this module.

**Non-Approved, Allowed Algorithms with No Security Claimed:**

| Name | Caveat | Use and Function |
|---|---|---|
| AES-KWP-PUFKEK | non-approved but allowed with no security claimed under IG 2.4.A when using PUFKEK to decrypt data. | AES Key Wrap with Padding (AES-KWP) applied on the TKH2, TKU2, or ZKU2 using PUFKEK as the AES key. |

Table 6: Non-Approved, Allowed Algorithms with No Security Claimed

**Non-Approved, Not Allowed Algorithms:**

N/A for this module.

## 2.6 Security Function Implementations

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Symmetric encryption with AES | BC-UnAuth | Perform encryption of data with AES | Key Size:128-, 192-, 256-bits<br>Security Strength:128-, 192- or 256-bits | AES-CBC: (A5631)<br>AES-ECB: (A5631) |
| Symmetric decryption with AES | BC-UnAuth | Perform decryption of data with AES | Key Size:128-, 192-, 256-bits<br>Security Strength:128-, 192- or 256-bits | AES-CBC: (A5631)<br>AES-ECB: (A5631) |
| Key wrapping with AES-KW or AES-KWP | KTS-Wrap | Perform encryption of data with AES-KW or AES-KWP in an authenticated mode | Standard:SP 800-38F<br>IG D.G:approved method from IG D.G<br>Caveat:Key establishment methodology provides between 128 and 256 bits of security strength | AES-KW: (A5631)<br>AES-KWP: (A5631) |
| Key unwrapping with AES- | KTS-Unwrap | Perform the unwrapping of a wrapped key | Standard:SP 800-38F<br>IG D.G:approved method from IG D.G<br>Caveat:Key establishment | AES-KW: (A5631)<br>AES-KWP: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| KW or AES-KWP | | with AES-KW or AES-KWP | methodology provides between 128 and 256 bits of security strength | |
| Authenticated symmetric encryption with AES-GCM | BC-Auth | Perform encryption of data with AES-GCM | Key Size:128-, 192-, 256-bits Security Strength:128-, 192- or 256-bits | AES-GCM: (A5631, A6119) |
| Authenticated symmetric decryption with AES-GCM | BC-Auth | Perform decryption of data with AES-GCM | Key Size:128-, 192-, 256-bits Security Strength:128-, 192- or 256-bits | AES-GCM: (A5631, A6119) |
| Key pair generation with ECDSA | AsymKeyPair-KeyGen CKG | Generate an ECDSA key pair | Curve:P-224, P-256, P-384, P-521 Security Strength:between 112 and 256 bits | ECDSA KeyGen (FIPS186-5): (A5631) CKG: () |
| Key verification with ECDSA | AsymKeyPair-KeyVer | Verify an ECDSA public key | Curve:P-224, P-256, P-384, P-521 Security Strength:between 112 and 256 bits | ECDSA KeyVer (FIPS186-5): (A5631) |
| Digital Signature generation with ECDSA | DigSig-SigGen | Generate a digital signature on a message with ECDSA | Curve:P-224, P-256, P-384, P-521 Security Strength:between 112 and 256 bits | ECDSA SigGen (FIPS186-5): (A5631) |
| Digital Signature verification with ECDSA | DigSig-SigVer | Verify a digital signature of a message with ECDSA | Curve:P-224, P-256, P-384, P-521 Security Strength:between 112 and 256 bits | ECDSA SigVer (FIPS186-5): (A5631) |
| Key pair generation with EDDSA | AsymKeyPair-KeyGen CKG | Generate an EDDSA key pair | Curve:Ed25519, Ed448 Security Strength:128 or 224 bits | EDDSA KeyGen: (A5631) CKG: () |
| Key verification with EDDSA | AsymKeyPair-KeyVer | Verify an EDDSA public key | Curve:Ed25519, Ed448 Security Strength:128 or 224 bits | EDDSA KeyVer: (A5631) |
| Digital Signature generation with EDDSA | DigSig-SigGen | Generate a digital signature on a message with EDDSA | Curve:Ed25519, Ed448 Security Strength:128 or 224 bits | EDDSA SigGen: (A5631) |
| Digital Signature verification with EDDSA | DigSig-SigVer | Verify a digital signature on a message signed with EDDSA | Curve:Ed25519, Ed448 Security Strength:128 or 224 bits | EDDSA SigVer: (A5631) |
| Random number generation | DRBG | Produce a sequence of random | Mode:SHA2-512 | Hash DRBG: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| with Hash DRBG | | numbers from Hash DRBG | | |
| Symmetric Key Generation (SP800-133rev2 section 4 example 1, B = U) | CKG | Generation of symmetric keys using Hash DRBG | | Hash DRBG: (A5631)<br>CKG: ()<br>Key Type: Symmetric |
| Symmetric Key Generation (SP800-133rev2 Section 6.3 method 2) | CKG | Exclusive-ORing two key components ( TKU0/TKH0/ZKU0 and TKU1/TKH1/ZKU1) and one data component ( TKU2/TKH2/ZKU2 ) | | Hash DRBG: (A5631)<br>CKG2: ()<br>Key Type: Symmetric |
| Message authentication code with HMAC | MAC | Generate a MAC of a message using HMAC | Key Size:8-8192 | HMAC-SHA-1: (A5631)<br>HMAC-SHA2-224: (A5631)<br>HMAC-SHA2-256: (A5631)<br>HMAC-SHA2-384: (A5631)<br>HMAC-SHA2-512: (A5631)<br>HMAC-SHA3-224: (A5631)<br>HMAC-SHA3-256: (A5631)<br>HMAC-SHA3-512: (A5631) |
| Elliptic curve Diffie- | KAS-Full | Establish a shared secret key using ECDH | IG:IG D.F Scenario 2, path (2), end-to-end<br>Key Confirmation:no | KAS-ECC Sp800- |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Hellman key agreement | | | Key derivation:KDA (tested as part of KAS certificate)<br>Caveat:Key establishment methodology provides between 112 and 256 bits of security strength | 56Ar3: (A5631) |
| Key derivation with KDA OneStep | KAS-56CKDF | Derive keying material from a secret value | Derived Key Length:256 bits<br>Auxiliary Functions Supported:SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-512 | KDA OneStep SP800-56Cr2: (A5631) |
| Key derivation with KDF ANS 9.63 | KAS-135KDF | Derive keying material from a secret value using the ANS X9.63-2001 derivation function | Derived Key Length:128-4096 bits<br>Hash Algorithms Supported:SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 | KDF ANS 9.63: (A5631) |
| Key derivation with KBKDF | KBKDF | Derive additional keying material from a pre-existing cryptographic key | KDF Mode:Counter<br>HMAC Supported:HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 | KDF SP800-108: (A5631) |
| Key encapsulation with RSA (KTS-Encapsulation) | KTS-Encap | Perform encapsulation of cryptographic keys with RSA | Standard:SP 800-56Brev2<br>IG D.G:approved method from IG D.G<br>Key confirmation:no<br>Caveat:Key establishment methodology provides between 112 and 200 bits of security strength<br>Modulo:2048, 3072, 4096 on all certs and 8192 only on Cert. #5631 | KTS-IFC: (A5631, A6410, A6409) |
| Key decapsulation with RSA (KTS-Decapsulation) | KTS-Decap | Perform decapsulation of cryptographic keys with RSA | Standard:SP 800-56Br2<br>IG D.G:approved method from IG D.G<br>Key confirmation:no<br>Caveat:Key establishment methodology provides between 112 and 200 bits of security strength | KTS-IFC: (A5631, A6409, A6410) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | Modulo:2048, 3072, 4096 on all certs and 8192 only on Cert. #5631 | |
| Key pair generation with LMS | AsymKeyPair-KeyGen CKG | Generate an LMS key pair | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | LMS KeyGen: (A5631) CKG: () SHA2-256: (A5631) SHAKE-256: (A5631) Hash DRBG: (A5631) |
| Digital Signature generation with LMS | DigSig-SigGen | Generate an LMS signature on a message | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, | LMS SigGen: (A5631) SHA2-256: (A5631) SHAKE-256: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | |
| Digital Signature verification with LMS | DigSig-SigVer | Verify a digital signature generated on a message signed with LMS | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, | LMS SigVer: (A5631) SHA2-256: (A5631) SHAKE-256: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | |
| Key pair generation with ML-DSA | AsymKeyPair-KeyGen CKG | Generate an ML-DSA key pair | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87 | ML-DSA KeyGen: (A5631) CKG: () SHAKE-256: (A5631) SHAKE-128: (A5631) Hash DRBG: (A5631) |
| Digital Signature generation with ML-DSA (External Interface, Pure, Hedged) | DigSig-SigGen | Generate an ML-DSA digital signature using the external signature interface in pure mode (full message input) in non-deterministic (hedged) mode. | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87 Signature Interface:external Pre Hash:pure Deterministic:No | ML-DSA SigGen: (A5631) SHAKE-256: (A5631) SHAKE-128: (A5631) Hash DRBG: (A5631) |
| Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, Message) | DigSig-SigGen | Generate an ML-DSA digital signature using the external signature interface in pre-hash mode in non-deterministic (hedged) mode, accepting the full message and computing | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87 Signature Interface:external Pre Hash:preHash Deterministic:No Hash Algorithms:SHA2-512 Message Input:Message | ML-DSA SigGen: (A5631) SHAKE-256: (A5631) SHAKE-128: (A5631) Hash DRBG: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | the digest internally with SHA2-512. | | SHA2-512: (A5631) |
| Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM) | DigSig-SigGen | Generate an ML-DSA digital signature using the external signature interface in pre-hash mode in non-deterministic (hedged) mode, accepting an externally computed message digest (PHM); the module performs the external-interface formatting over the supplied digest. | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Signature Interface:external<br>Pre Hash:preHash<br>Deterministic:No<br>Hash Algorithms:SHA2-512<br>Message Input:PHM (pre-hashed message digest) | ML-DSA SigGen: (A5631)<br>SHAKE-256: (A5631)<br>SHAKE-128: (A5631)<br>Hash DRBG: (A5631)<br>SHA2-512: (A5631) |
| Digital Signature generation with ML-DSA (Internal Interface, Hedged, External Mu) | DigSig-SigGen | Generate an ML-DSA digital signature using the internal signature interface in non-deterministic (hedged) mode, accepting an externally computed mu value. | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Signature Interface:internal<br>Deterministic:No<br>External Mu:Yes | ML-DSA SigGen: (A5631)<br>SHAKE-256: (A5631)<br>SHAKE-128: (A5631)<br>Hash DRBG: (A5631) |
| Digital Signature verification with ML-DSA (External Interface, Pure) | DigSig-SigVer | Verify an ML-DSA signature using the external signature interface in pure mode (full message input). | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Signature Interface:external<br>Pre Hash:pure | ML-DSA SigVer: (A5631)<br>SHAKE-256: (A5631)<br>SHAKE-128: (A5631) |
| Digital Signature | DigSig-SigVer | Verify an ML-DSA signature | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87 | ML-DSA SigVer: |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| verification with ML-DSA (External Interface, PreHash, Message) | | using the external signature interface in pre-hash mode, accepting the full message and computing the digest internally with SHA2-512. | Signature Interface:external<br>Pre Hash:preHash<br>Hash Algorithms:SHA2-512<br>Message Input:Message | (A5631) SHAKE-256:<br>(A5631) SHAKE-128:<br>(A5631) SHA2-512:<br>(A5631) |
| Digital Signature verification with ML-DSA (External Interface, PreHash, PHM) | DigSig-SigVer | Verify an ML-DSA signature using the external signature interface in pre-hash mode, accepting an externally computed message digest (PHM); the module performs the external-interface formatting over the supplied digest. | Parameters:ML-DSA-44, ML-DSA-65, ML-DSA-87<br>Signature Interface:external<br>Pre Hash:preHash<br>Hash Algorithms:SHA2-512<br>Message Input:PHM (pre-hashed message digest) | ML-DSA SigVer:<br>(A5631) SHAKE-256:<br>(A5631) SHAKE-128:<br>(A5631) SHA2-512:<br>(A5631) |
| Key pair generation with ML-KEM | AsymKeyPair-KeyGen CKG | Generate an ML-KEM key pair | Parameters:ML-KEM-512, ML-KEM-768, ML-KEM-1024<br>IG:D.S Scenario 1 | ML-KEM KeyGen:<br>(A5631) CKG: ()<br>SHA3-256:<br>(A5631) SHA3-512:<br>(A5631) SHAKE-256:<br>(A5631) SHAKE-128:<br>(A5631) Hash DRBG:<br>(A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Key encapsulation with ML-KEM | KEM-Encap | KEM encapsulation with ML-KEM: the module performs key encapsulation using operator-provided inputs, per IG D.S Scenario 1. | Parameters:ML-KEM-512, ML-KEM-768, ML-KEM-1024<br>IG:IG D.S Scenario 1 | ML-KEM EncapDecap: (A5631)<br>SHA3-256: (A5631)<br>SHA3-512: (A5631)<br>SHAKE-256: (A5631)<br>SHAKE-128: (A5631)<br>Hash DRBG: (A5631) |
| Key decapsulation with ML-KEM | KEM-Decap | KEM decapsulation with ML-KEM: the module performs key decapsulation using operator-provided inputs, per IG D.S Scenario 1. | Parameters:ML-KEM-512, ML-KEM-768, ML-KEM-1024<br>IG:IG D.S Scenario 1 | ML-KEM EncapDecap: (A5631)<br>SHA3-256: (A5631)<br>SHA3-512: (A5631)<br>SHAKE-256: (A5631)<br>SHAKE-128: (A5631) |
| Key pair generation with RSA | AsymKeyPair-KeyGen<br>CKG | Generate an RSA key pair | Modulus:2048, 3072, 4096, 8192<br>Security Strength:between 112 and 192 bits | RSA KeyGen (FIPS186-5): (A5631)<br>CKG: () |
| RSA Signature Primitive 2.0 | DigSig-SigGen | Generate a digital signature on a message with an existing message digest using RSA | Modulus:2048, 3072, 4096<br>Security Strength:between 112 and 150 bits | RSA Signature Primitive: (A5631, A6409, A6410) |
| Digital Signature generation with RSA | DigSig-SigGen | Generate a digital signature on a message with RSA | Modulus:2048, 3072, 4096<br>Security Strength:between 112 and 150 bits | RSA SigGen (FIPS186-5): (A5631, A6409, A6410) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Digital Signature verification with RSA | DigSig-SigVer | Verify a digital signature on a message signed with RSA | Modulus:2048, 3072, 4096 Security Strength:between 112 and 150 bits | RSA SigVer (FIPS186-5): (A5631, A6409, A6410) |
| Compute message digest with SHA | SHA | Generate a hash digest of a message with SHA | | SHA-1: (A5631) SHA2-224: (A5631) SHA2-256: (A5631) SHA2-384: (A5631) SHA2-512: (A5631) SHA2-512/224: (A5631) SHA2-512/256: (A5631) SHA3-224: (A5631) SHA3-256: (A5631) SHA3-384: (A5631) SHA3-512: (A5631) |
| Extendable output functions | XOF | Generate a customizable length hash digest on a message using an extendable output function | | SHAKE-128: (A5631) SHAKE-256: (A5631) |
| Key pair generation with SLH-DSA | AsymKeyPair-KeyGen CKG | Generate an SLH-DSA key pair | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | SLH-DSA KeyGen: (A5631) CKG: () SHA2-256: (A5631) SHA2-512: (A5631) SHAKE-128: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | | SHAKE-256: (A5631) HMAC-SHA2-256: (A5631) HMAC-SHA2-512: (A5631) Hash DRBG: (A5631) |
| Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged) | DigSig-SigGen | Generate an SLH-DSA digital signature using the external signature interface in pure mode (full message input) in non-deterministic (hedged) mode. | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s Signature Interface:external Pre Hash:pure Deterministic:No | SLH-DSA SigGen: (A5631) SHA2-256: (A5631) SHA2-512: (A5631) SHAKE-128: (A5631) SHAKE-256: (A5631) HMAC-SHA2-256: (A5631) HMAC-SHA2-512: (A5631) Hash DRBG: (A5631) |
| Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message) | DigSig-SigGen | Generate an SLH-DSA digital signature using the external signature interface in pre-hash mode in non-deterministic (hedged) mode, accepting the full message and computing the digest | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s Signature Interface:external Pre Hash:preHash Deterministic:No | SLH-DSA SigGen: (A5631) SHA2-256: (A5631) SHA2-512: (A5631) SHAKE-128: (A5631) SHAKE-256: (A5631) HMAC-SHA2-256: |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | internally with SHA2-512. | Hash Algorithms:SHA2-512<br>Message Input:Message | (A5631)<br>HMAC-SHA2-512:<br>(A5631)<br>Hash DRBG:<br>(A5631) |
| Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM) | DigSig-SigGen | Generate an SLH-DSA digital signature using the external signature interface in pre-hash mode in non-deterministic (hedged) mode, accepting an externally computed message digest (PHM); the module performs the external-interface formatting over the supplied digest. | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Signature Interface:external<br>Pre Hash:preHash<br>Deterministic:No<br>Hash Algorithms:SHA2-512<br>Message Input:PHM (pre-hashed message digest) | SLH-DSA SigGen:<br>(A5631)<br>SHA2-256:<br>(A5631)<br>SHA2-512:<br>(A5631)<br>SHAKE-128:<br>(A5631)<br>SHAKE-256:<br>(A5631)<br>HMAC-SHA2-256:<br>(A5631)<br>HMAC-SHA2-512:<br>(A5631)<br>Hash DRBG:<br>(A5631) |
| Digital Signature verification with SLH-DSA (External Interface, Pure) | DigSig-SigVer | Verify an SLH-DSA signature using the external signature interface in pure mode (full message input). | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Signature Interface:external<br>Pre Hash:pure | SLH-DSA SigVer:<br>(A5631)<br>SHA2-256:<br>(A5631)<br>SHA2-512:<br>(A5631)<br>SHAKE-128:<br>(A5631)<br>SHAKE-256:<br>(A5631)<br>HMAC-SHA2-256:<br>(A5631)<br>HMAC-SHA2-512:<br>(A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Digital Signature verification with SLH-DSA (External Interface, PreHash, Message) | DigSig-SigVer | Verify an SLH-DSA signature using the external signature interface in pre-hash mode, accepting the full message and computing the digest internally with SHA2-512. | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Signature Interface:external<br>Pre Hash:preHash<br>Hash Algorithms:SHA2-512<br>Message Input:Message | SLH-DSA SigVer: (A5631)<br>SHA2-256: (A5631)<br>SHA2-512: (A5631)<br>SHAKE-128: (A5631)<br>SHAKE-256: (A5631)<br>HMAC-SHA2-256: (A5631)<br>HMAC-SHA2-512: (A5631) |
| Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) | DigSig-SigVer | Verify an SLH-DSA signature using the external signature interface in pre-hash mode, accepting an externally computed message digest (PHM); the module performs the external-interface formatting over the supplied digest. | Parameters:SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s<br>Signature Interface:external<br>Pre Hash:preHash<br>Hash Algorithms:SHA2-512<br>Message Input:PHM (pre-hashed message digest) | SLH-DSA SigVer: (A5631)<br>SHA2-256: (A5631)<br>SHA2-512: (A5631)<br>SHAKE-128: (A5631)<br>SHAKE-256: (A5631)<br>HMAC-SHA2-256: (A5631)<br>HMAC-SHA2-512: (A5631) |
| Digital Signature generation with HSS | DigSig-SigGen | Generate a digital signature on a message with HSS | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, | HSS SigGen: ()<br>SHA2-256: (A5631)<br>SHAKE-256: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | |
| Digital Signature verification with HSS | DigSig-SigVer | Verify a digital signature on a message with HSS | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, | HSS SigVer: () SHA2-256: (A5631) SHAKE-256: (A5631) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | |
| Key pair generation with HSS | AsymKeyPair-KeyGen CKG | Generate a HSS key pair | LMOTS Modes:LMOTS_SHA256_N24_W2, LMOTS_SHA256_N24_W4, LMOTS_SHA256_N24_W8, LMOTS_SHAKE_N24_W2, LMOTS_SHAKE_N24_W4, LMOTS_SHAKE_N24_W8, LMOTS_SHA256_N32_W2, LMOTS_SHA256_N32_W4, LMOTS_SHA256_N32_W8, LMOTS_SHAKE_N32_W2, LMOTS_SHAKE_N32_W4, LMOTS_SHAKE_N32_W8 LMS Modes:LMS_SHA256_M24_H5, LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHAKE_M24_H5, LMS_SHAKE_M24_H10, LMS_SHAKE_M24_H15, LMS_SHAKE_M24_H20, LMS_SHAKE_M24_H25, LMS_SHA256_M32_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHAKE_M32_H5, LMS_SHAKE_M32_H10, LMS_SHAKE_M32_H15, LMS_SHAKE_M32_H20, LMS_SHAKE_M32_H25 | HSS KeyGen: () CKG: () SHA2-256: (A5631) SHAKE-256: (A5631) Hash DRBG: (A5631) |
| Key derivation with SP800-56Cr2 | KAS-56CKDF | Derive keying material from secret material | Hash Algorithms Supported:SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2- | KDA OneStep SP800- |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | with SP800-56Cr2 KDF | 512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 | 56Cr2: (A5631) |

Table 7: Security Function Implementations

## 2.7 Algorithm Specific Information

**AES-GCM**

The implements AES-GCM compliant with IG C.H scenario 2, such that the IV is generated entirely randomly within the module. The generation uses an Approved DRBG that is internal to the module's boundary, and the IV length of 96 bits.

**KAS-ECC**

The module implements key agreement compliant to IG D.F following scenario 2 path (2). The key agreement scheme used is One-Pass Diffie-Hellman scheme (1e, 1s, ECDH) where the module's key is static and the other party acts as the Initiator that provides the ephemeral key. The CAVP testing was performed end-to-end with a KAS-ECC CAVP certificate issued (without key confirmation).

The following is a list of applicable sections from SP800-56Arev3 section 5.6.2 and how the respective shall requirements are met.

- Assurance that the key pair was generated correctly is met as specified 5.6.2.1.1 item a. since the module generates the key pair for ECDH using NIST curves from SP 800-186.
- Assurance of private key validity is met as specified 5.6.2.1.2 item a. since the owner uses the module generates the key pair as specified above.
- Assurance of public key validity is met as specified 5.6.2.1.3 item a. since the module performs full public key validation.
- Assurance of pair-wise consistency is met as specified 5.6.2.1.4 item a. since the module generates the key pair as specified above.
- Assurance of possession of the private key is met as specified 5.6.2.1.5 item a. since the module generates the key pair as specified above.
- Assurance of public-key validity is met as specified in 5.6.2.2.2 since module performs public key validation on the received public key.
- Assurance of private-key possession: As explained above the module only allows use of ephemeral keys from other peer so assurance of private-key possession is not mandatory when using ephemeral keys.

**SHA-1**

The module only offers services with approved usage of SHA-1 as part of Message Digest or Message Authentication Code. The module does not offer any service with non-approved usage for SHA-1 such as Digital Signature Generation. The use of SHA-1 by the *Hash* service, using the *Compute Message digest with SHA* security function, is only approved for integrity verification purposes and non-approved if used as part of a digital signature generation.

**RSA**

The module's RSA signature generation and signature verification implementation were CAVP validated with moduli sizes 2048, 3072, 4096. As allowed by IG C.F, the module additionally

implements signature generation and signature verification using an approved 8192-bit modulus for which there is currently no CAVP testing available.

**ML-KEM**

The module does not establish SSPs using an approved key encapsulation mechanism (KEM). However, it does offer some or all of the underlying KEM cryptographic functionality to be used by an external operator/application as part of an approved KEM.

When using an externally obtained ML-KEM public encapsulation key, the authenticated operator shall ensure that the ownership of the key has been established prior to invoking the ML-KEM encapsulation service. This assurance may be obtained using a trusted third party (e.g., a certificate authority) or another mechanism appropriate for the deployment, consistent with the guidance of SP 800-227 Section 4.5.

## 2.8 RBG and Entropy

| Cert Number | Vendor Name |
|---|---|
| E96 | Crypto4A Technologies, Inc. |

Table 8: Entropy Certificates

| Name | Type | Operational Environment | Sample Size | Entropy per Sample | Conditioning Component |
|---|---|---|---|---|---|
| Crypto4A QASM Entropy Source | Physical | QASM 1.1 (Xilinx Zynq UltraScale+) | 256 bit | 239.87 bits | N/A |

Table 9: Entropy Sources

## 2.9 Key Generation

The module claims CKG for symmetric and asymmetric keys. Both Asymmetric and Symmetric CKG are in compliance with section 4, example 1 of SP800-133rev2. In addition, the module implements CKG for symmetric keys (considered data) compliant to section 6.3 method 2. TKU and TKH are derived using XOR of three components that are generated using module's SP 800-90A DRBG algorithm. The components TKU0/TKH0 and TKU1/TKH1 are mapping to the key components following SP800-133r2 section 6.3 bullet 2 where both exclusively provide the required security strength of the resulting key. The components TKU2/TKH2 are mapping to data following SP800-133r2 section 6.2 bullet 2 which does not contribute to the security strength of the resulting key

## 2.10 Key Establishment

The module can be used to perform KAS-full through a SP800-56Arev3-compliant scheme that conforms to IG D.F. The onePassDH scheme is used with the P-224, P-256, P-384, and P-521 curves. Key confirmation is not implemented.

The module can be used to perform KTS through a SP800-56Brev2-compliant scheme that conforms to IG D.G. The KTS-OAEP-basic scheme is used with the 2048, 3072, 4096, and 8192 modulo. Key encapsulation and decapsulation are both implemented.

The module can be used to perform KTS through SP 800-38F compliant scheme that conforms to IG D.G. The AES-KW and AES-KWP are used with 128-, 192-, and 256-bit keys.

## 2.11 Industry Protocols

The module implements an ANS X9.63 KDF as specified in SP 800-135rev1 and this KDF has been validated by the CAVP. No parts of the ANSI X9.63-2001 protocol, other than the KDF, have been tested by the CAVP or the CMVP.

# 3 Cryptographic Module Interfaces

## 3.1 Ports and Interfaces

The main interfaces implemented by the AMI are: a Universal Asynchronous Receiver-Transmitter (UART) interface, four Universal Serial Bus (USB) version 3.0 interfaces and six Peripheral Component Interconnect express (PCIe) dual lane buses that allow the QASM to interact with external system management support, the single board computers (SBCs), and other external interfaces. The module does not implement a control output interface.

| Physical Port | Logical Interface(s) | Data That Passes |
|---|---|---|
| AMI-Connector<br>- 6x 2-lane PCIe<br>- 4x USB 3.0<br>- 2x I2C<br>- 1x UART | Data Input | All data provided to the QASM as part of the data processing commands. |
| AMI-Connector<br>- 6x 2-lane PCIe<br>- 4x USB 3.0<br>- 2x I2C<br>- 1x UART | Data Output | All data provided by the QASM as part of the data processing commands. |
| AMI-Connector<br>- 6x 2-lane PCIe<br>- 4x USB 3.0<br>- 1x REFCLKIN<br>- 1x PPSIN<br>- 2x I2C<br>- 1x UART<br>- 24x GPIO | Control Input | All control provided to the QASM as part of the data processing commands. GPIO pins can be used to trigger manual zeroization |
| AMI-Connector<br>- 6x 2-lane PCIe<br>- 4x USB 3.0<br>- 2x I2C<br>- 1x UART<br>- 6x GPIO | Status Output | Status info via return codes implemented through different protocols and status indicator signals through th GPIO pins. |
| AMI-Connector<br>- 5V<br>- 12V | Power | None |

Table 10: Ports and Interfaces

# 4 Roles, Services, and Authentication

## 4.1 Authentication Methods

| Method Name | Description | Security Mechanism | Strength Each Attempt | Strength per Minute |
|---|---|---|---|---|
| ECDSA Signature | Identity based authentication for the Owner, Security Officer and Access User roles using ECDSA digital signature verification. | ECDSA Digital Signatures | The strength of this authentication mechanism is judged using ECDSA P-224, though all curve sizes may be used for authentication. According to NIST's SP800-57 Part 1 Table 2, the security strength of ECDSA P-224 is 112 bits. An attacker would therefore need to guess the user's key to defeat the QASM authentication method. Each guess would have a probability of 1/2^112 of success. | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^112 which is less than 1/100,00 set as the strength objective. |
| RSA Signature | Identity based authentication for the Owner, Security Officer and Access User roles using RSA digital signature verification. | RSA Digital Signatures | The strength of this authentication mechanism is judged using 2048-bit RSA, though all key sizes may be used for authentication. According to NIST's SP800-57 Part 1 Table 2, the security strength of 2048-bit RSA is 112 bits. An attacker would therefore need to guess the user's key to defeat the QASM authentication method. Each guess would have a probability of 1/2^112 of success. | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^112 which is less than 1/100,00 set as the strength objective. |
| EdDSA Signature | Identity based authentication for the Owner, Security Officer and Access User roles using EdDSA digital signature verification. | EdDSA Signature | The strength of this authentication mechanism is judged using EdDSA Ed25519, though all curve sizes may be used for authentication. According to NIST's FIPS 186-5 Section 7.1, the security strength of EdDSA Ed25519 is 128 bits. An | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * |

| Method Name | Description | Security Mechanism | Strength Each Attempt | Strength per Minute |
|---|---|---|---|---|
| | | | attacker would therefore need to guess the user's key to defeat the QASM authentication method. Each guess would have a probability of 1/2^128 of success. | 1/2^128 which is less than 1/100,00 set as the strength objective. |
| HSS Signature | Identity based authentication for the Owner, Security Officer and Access User roles using HSS digital signature verification. | HSS Signature | The strength of this authentication mechanism is based on HSS using SHA-256/SHAKE256. Although all supported HSS parameter sets may be used for authentication, the authentication strength is determined by the lowest security strength parameter set supported by the module (M = 24, N = 24). Therefore, the probability of a successful random authentication attempt is 1/2^192 per attempt, which is well below the threshold set by the vendor of 1/10^6. | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^192 which is less than 1/100,00 set as the strength objective. |
| ML-DSA Signature | Identity based authentication for the Owner, Security Officer and Access User roles using ML-DSA digital signature verification. | ML-DSA Signature | The strength of this authentication mechanism is judged using ML-DSA-44, though all key sizes may be used for authentication. According to NIST's FIPS 204 Section 4, the security strength of ML-DSA-44 is 128 bits. An attacker would therefore need to guess the user's key to defeat the QASM authentication method. Each guess would have a probability of 1/2^128 of success, which is well below the threshold set by the vendor of 1/10^6. | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^128 which is less than 1/100,00 set as the strength objective. |
| LMS Signature | Identity based authentication for the Owner, Security Officer | LMS Signature | The strength of this authentication mechanism is based on LMS using SHA-256/SHAKE256. | The module can process at most 2^16 attempts per minute (i.e. 65536 |

| Method Name | Description | Security Mechanism | Strength Each Attempt | Strength per Minute |
|---|---|---|---|---|
| | and Access User roles using LMS digital signature verification. | | Although all supported LMS parameter sets may be used for authentication, the authentication strength is determined by the lowest security strength parameter set supported by the module (M = 24, N = 24). Therefore, the probability of a successful random authentication attempt is 1/2^192 per attempt, which is well below the threshold set by the vendor of 1/10^6. | attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^192 which is less than 1/100,00 set as the strength objective. |
| SLH-DSA Signature | Identity based authentication for the Owner, Security Officer and Access User roles using SLH-DSA digital signature verification. | SLH-DSA Signature | The strength of this authentication mechanism is judged using SLH-DSA SHA128/SHAKE128, though all key sizes may be used for authentication. According to NIST's FIPS 205 Section 11, the security strength of SLH-DSA SHA128/SHAKE128 is 128 bits. An attacker would therefore need to guess the user's key to defeat the QASM authentication method. Each guess would have a probability of 1/2^128 of success, which is well below the threshold set by the vendor of 1/10^6. | The module can process at most 2^16 attempts per minute (i.e. 65536 attempts) which will yield probability of successful authentication in a minute <= 65536 * 1/2^128 which is less than 1/100,00 set as the strength objective. |

Table 11: Authentication Methods

The QASM identity-based authorization process has two steps. The first step is to prepare a request containing information about the service to be performed. The second step is to sign the request with the appropriate authorities and submit the signature(s). The QASM will perform the service once the signature(s) are submitted and validated.

An operator can choose an identity to authenticate into by signing an authorization request to login with the private key corresponding to that identity during normal operation. To authenticate into a different identity (and corresponding role) an operator would need to logout and login again with the desired identity’s private key.

The Owner is the only role that is pre-installed with Crypto4A's ECDSA P-384 public key. When a new owner of the physical device installs the platform, they have the option to configure a new Owner. The resident Owner must sign the new Owner's public key and submit a request to the QASM to install the new Owner's public key i.e., *Take ownership*. The Owner role (as well as the Security Officer and Access User) can use any approved signature scheme to authenticate the owner (i.e., ECDSA, EdDSA, HSS, LMS, ML-DSA, RSA, SLH-DSA). The module accepts a public key that respects the list of approved mechanisms. The Owner role can create a Security Officer and the Access User. The Security Officer can create an Access User. An Access User cannot create itself.

The authentication strength objective of the module is 1/1,000,000, which corresponds to a false acceptance rate of 1/100,000. The smallest key strength supported by the module is 112 bits. The chance of a random authentication attempt falsely succeeding is then 1/2^112, which is less than the claimed strength objective of 1/1,000,000. Only 2^16 attempts per minute are allowed, therefore the probability of successful authentication in a minute is less than or equal to 2^16 * 1/2^112, which is less than the false acceptance rate of 1/100,000.

If the module is rebooted, the operator will need to re-authenticate to the module.

## 4.2 Roles

| Name | Type | Operator Type | Authentication Methods |
|---|---|---|---|
| Owner | Identity | Crypto Officer | ECDSA Signature<br>RSA Signature<br>EdDSA Signature<br>HSS Signature<br>ML-DSA Signature<br>LMS Signature<br>SLH-DSA Signature |
| Security Officer | Identity | Crypto Officer | ECDSA Signature<br>RSA Signature<br>EdDSA Signature<br>HSS Signature<br>ML-DSA Signature<br>LMS Signature<br>SLH-DSA Signature |
| Access User | Identity | User | ECDSA Signature<br>RSA Signature<br>EdDSA Signature<br>HSS Signature<br>ML-DSA Signature<br>LMS Signature<br>SLH-DSA Signature |

Table 12: Roles

The QASM supports two CO roles and multiple user roles. The maintenance role is not supported. The module does not support concurrent operators.

- Owner: A crypto officer role. The owner oversees the operation of the module (e.g. updating firmware and zeroizing the module). The owner can give up ownership of its

role to a different owner through a secure owner authority transfer .There is a single owner permitted per module.

- Security Officer: A crypto officer role. There is a single security officer per module. The security officer manages the access policy and oversees adding and removing users. The owner can change the security officer through a secure security officer authority transfer.
- Access User: A user role. The access user is authenticated, and can perform a range of services such as key generation, hashing, encryption and decryption, key derivation, etc.

## 4.3 Approved Services

The module only supports approved services. For all approved security services, a global module-level indicator is used via a successful completion of the service following example 2 of IG 2.4.C. When a service completes successfully, “HSM_RET_OK” is returned.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Generate random data | Provide DRBG output | HSM_RET_OK | number of bits | random numbers | Random number generation with Hash DRBG | Owner<br>- DRBG Entropy Input: G,E,Z<br>- DRBG V: G,W,E<br>- DRBG C: G,W,E<br>- DRBG Seed: G,E,Z<br>Security Officer<br>- DRBG Entropy Input: G,E,Z<br>- DRBG V: G,W,E<br>- DRBG C: G,W,E<br>- DRBG Seed: G,E,Z<br>Access User<br>- DRBG Entropy Input: G,E,Z<br>- DRBG V: G,W,E<br>- DRBG C: |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | G,W,E<br>- DRBG Seed: G,E,Z |
| Key Derivation (KBKDF) | Perform KBKDF key derivation | HSM_RET_OK | Key Derivation Key | KBKDF Derived Key | Key derivation with KBKDF<br>Key verification with ECDSA<br>Key verification with EDDSA | Owner<br>- KBKDF Key Derivation Keys (Generated): E<br>- KBKDF Key Derivation Keys (Established): W,E<br>- KBKDF Derived Key: G,R<br>Security Officer<br>- KBKDF Key Derivation Keys (Generated): E<br>- KBKDF Key Derivation Keys (Established): W,E<br>- KBKDF Derived Key: G,R<br>Access User<br>- KBKDF Key Derivation Keys (Generated): E<br>- KBKDF Key Derivation |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Keys (Established): E<br>- KBKDF Derived Key: G |
| Key Derivation (ANS 9.63) | Perform ANS 9.63 key derivation | HSM_RET_OK | Shared Secret | ANS X9.63 Derived Key | Key derivation with KDF ANS 9.63 | Owner<br>- KDF ANS 9.63 Derived Key: G<br>Security Officer<br>- KDF ANS 9.63 Derived Key: G<br>Access User<br>- KDF ANS 9.63 Derived Key: G |
| Key Derivation (SP800-56Cr2) | Perform SP800-56Cr2 key derivation | HSM_RET_OK | Shared Secret | KDA Derived Key | Key derivation with SP800-56Cr2 Key derivation with KDA OneStep | Owner<br>- SP800-56Cr2 Derived Key: G<br>Security Officer<br>- SP800-56Cr2 Derived Key: G<br>Access User<br>- SP800-56Cr2 Derived Key: G |
| Key agreement (SP800-56ARev3) | Perform ECDH key agreement | HSM_RET_OK | ECDH recipient public key | ECDH provider public key | Elliptic curve Diffie-Hellman key agreement | Owner<br>- ECDH Derived Key: G<br>- ECC Shared Secret: |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Key verification with ECDSA Key verification with EDDSA | E,Z,G<br>- ECDH Recipient Public Key: R,E,Z<br>- ECDH Provider Private Key: G,E,Z<br>- QASM_KE_ECC (Private Key): E<br>- QASM_KE_ECC (Public Key): E<br>- QASM_AA_ECC (Private Key): E<br>- QASM_AA_ECC (Public Key): E<br>Security Officer<br>- ECDH Derived Key: G<br>- ECC Shared Secret: E,Z,G<br>- ECDH Recipient Public Key: R,E,Z<br>- ECDH Provider Private Key: G,E,Z<br>- QASM_KE |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | _ECC (Private Key): E - QASM_KE _ECC (Public Key): E - QASM_AA _ECC (Private Key): E - QASM_AA _ECC (Public Key): E Access User - ECDH Derived Key: G - ECC Shared Secret: E,Z,G - ECDH Recipient Public Key: R,E,Z - ECDH Provider Private Key: G,E,Z - QASM_KE _ECC (Private Key): E - QASM_KE _ECC (Public Key): E - QASM_AA _ECC |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Private Key): E<br>- QASM_AA_ECC (Public Key): E |
| AES key generation | Generate a symmetric key | HSM_RET_OK | Key size | AES key | Symmetric Key Generation (SP800-133rev2 section 4 example 1, B = U) | Owner<br>- AES Symmetric Keys (Generated): G<br>- AES-GCM Symmetric Keys (Generated): G<br>- AES-KW Symmetric Keys (Generated): G<br>Security Officer<br>- AES Symmetric Keys (Generated): G<br>- AES-GCM Symmetric Keys (Generated): G<br>- AES-KW Symmetric Keys (Generated): G<br>Access User<br>- AES Symmetric Keys (Generated): G<br>- AES-GCM |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Symmetric Keys (Generated): G<br>- AES-KW Symmetric Keys (Generated): G |
| RSA key pair generation | Generate an RSA key pair | HSM_RET_OK | Modulus size | RSA Key Pair | Key pair generation with RSA Random number generation with Hash DRBG | Owner<br>- RSA Digital Signature Private Keys (Generated): G<br>- RSA Digital Signature Public Keys (Generated): G<br>Security Officer<br>- RSA Digital Signature Private Keys (Generated): G<br>- RSA Digital Signature Public Keys (Generated): G<br>Access User<br>- RSA Digital Signature Private Keys (Generated): G<br>- RSA |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Public Keys (Generated): G |
| ECDSA key pair generation | Generate an ECDSA key pair | HSM_RET_OK | Curve size | ECDSA Key Pair | Key pair generation with ECDSA<br>Random number generation with Hash DRBG | Owner<br>- ECDSA Digital Signature Private Keys (Generated): G<br>- ECDSA Digital Signature Public Keys (Generated): G<br>Security Officer<br>- ECDSA Digital Signature Private Keys (Generated): G<br>- ECDSA Digital Signature Public Keys (Generated): G<br>Access User<br>- ECDSA Digital Signature Private Keys (Generated): G<br>- ECDSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): G |
| EDDSA key pair generation | Generate an EDDSA key pair | HSM_RET_OK | Curve size | EdDSA Key Pair | Key pair generation with EDDSA Random number generation with Hash DRBG | Owner<br>- EDDSA Digital Signature Private Keys (Generated): G<br>- EDDSA Digital Signature Public Keys (generated): G<br>Security Officer<br>- EDDSA Digital Signature Private Keys (Generated): G<br>- EDDSA Digital Signature Public Keys (generated): G<br>Access User<br>- EDDSA Digital Signature Private Keys (Generated): G<br>- EDDSA Digital Signature Public Keys (generated): G |
| LMS key pair generation | Generate an LMS key pair | HSM_RET_OK | Parameter set | LMS Public Key | Key pair generatio | Owner<br>- LMS |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | n with LMS Key pair generation with HSS Random number generation with Hash DRBG | Digital Signature Private Keys (Generated): G<br>- LMS Digital Signature Public Keys (Generated): G,R Security Officer<br>- LMS Digital Signature Private Keys (Generated): G<br>- LMS Digital Signature Public Keys (Generated): G,R Access User<br>- LMS Digital Signature Private Keys (Generated): G<br>- LMS Digital Signature Public Keys (Generated): G,R |
| ML-DSA key pair generation | Generate an ML-DSA key pair | HSM_RET_OK | Parameter set | ML-DSA Public Key, ML-DSA Private Key ID | Key pair generation with ML-DSA Random number | Owner<br>- ML-DSA Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | generation with Hash DRBG | (Generated): G<br>- ML-DSA Digital Signature Public Keys (Generated): G<br>- ML-DSA Seed: G,E<br>Security Officer<br>- ML-DSA Digital Signature Private Keys (Generated): G<br>- ML-DSA Digital Signature Public Keys (Generated): G<br>- ML-DSA Seed: G,E<br>Access User<br>- ML-DSA Digital Signature Private Keys (Generated): G<br>- ML-DSA Digital Signature Public Keys (Generated): G<br>- ML-DSA Seed: G,E |
| ML-KEM key pair generation | Generate an ML-KEM key pair | HSM_RET_OK | Parameter set | ML-KEM public key ek | Key pair generation with ML-KEM | Owner<br>- ML-KEM Key Decapsulati |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Random number generation with Hash DRBG | on Private Key (Generated): G<br>- ML-KEM Key Encapsulation Public Key (Generated): G,R<br>- ML-KEM Seed: G,E<br>Security Officer<br>- ML-KEM Key Decapsulation Private Key (Generated): G<br>- ML-KEM Key Encapsulation Public Key (Generated): G,R<br>- ML-KEM Seed: G,E<br>Access User<br>- ML-KEM Key Decapsulation Private Key (Generated): G<br>- ML-KEM Key Encapsulation Public Key (Generated): G,R |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - ML-KEM Seed: G,E |
| SLH-DSA key pair generation | Generate an SLH-DSA key pair | HSM_RET_OK | Parameter set | SLH-DSA Key Pair | Key pair generation with SLH-DSA<br>Random number generation with Hash DRBG | Owner<br>- SLH-DSA Digital Signature Private Keys (Generated): G<br>- SLH-DSA Digital Signature Public Keys (Generated): G,R<br>- SLH-DSA Seed: G,E<br>Security Officer<br>- SLH-DSA Digital Signature Private Keys (Generated): G<br>- SLH-DSA Digital Signature Public Keys (Generated): G,R<br>- SLH-DSA Seed: G,E<br>Access User<br>- SLH-DSA Digital Signature Private Keys (Generated): G<br>- SLH-DSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): G,R<br>- SLH-DSA Seed: G,E |
| Hash | Generate a hash digest of a message | HSM_RET_OK | Message | Message Digest | Compute message digest with SHA | Owner<br>Security Officer<br>Access User |
| Keyed-hash | Generated a MAC from a message | HSM_RET_OK | Key, Message | MAC Tag | Message authentication code with HMAC | Owner<br>- HMAC Symmetric Keys (Established): E<br>Security Officer<br>- HMAC Symmetric Keys (Established): E<br>Access User<br>- HMAC Symmetric Keys (Established): E |
| Symmetric encryption/decryption | Perform symmetric encryption/decryption | HSM_RET_OK | AES Key, Plaintext Data | Ciphertext | Symmetric encryption with AES<br>Symmetric decryption with AES | Owner<br>- AES Symmetric Keys (Generated): E<br>- AES Symmetric Keys (Established): E<br>Security Officer<br>- AES Symmetric Keys (Generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - AES Symmetric Keys (Established): E<br>Access User<br>- AES Symmetric Keys (Generated): E<br>- AES Symmetric Keys (Established): E |
| Authenticated Symmetric encryption/decryption | Perform authenticated symmetric encryption/decryption | HSM_RET_OK | AES Key, Plaintext Data/Key to be wrapped | Ciphertext | Authenticated symmetric encryption with AES-GCM Authenticated symmetric decryption with AES-GCM | Owner<br>- AES-GCM Symmetric Keys (Generated): E<br>- AES-GCM Symmetric Keys (Established): E<br>- TKA Wrapping Key for ATP0 (TKAWK0): E<br>- TKA Wrapping Key for ATP1 (TKAWK1): E<br>- Tenant Master Key (TMK): E<br>Security Officer<br>- AES-GCM Symmetric Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E<br>- AES-GCM Symmetric Keys (Established): E<br>- TKA Wrapping Key for ATP0 (TKAWK0): E<br>- TKA Wrapping Key for ATP1 (TKAWK1): E<br>Access User<br>- AES-GCM Symmetric Keys (Generated): E<br>- AES-GCM Symmetric Keys (Established): E |
| ECDSA Signature generation | Generate an ECDSA digital signature | HSM_RET_OK | Private Key, Message, Hash Algorithm | Digital Signature | Digital Signature generation with ECDSA | Owner<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- QASM_IA_ |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ECC (Private Key): E<br>- QASM_SCA_COM (Private Key): E<br>Security Officer<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- QASM_IA_ECC (Private Key): E<br>- QASM_SCA_COM (Private Key): E<br>Access User<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E<br>- QASM_IA_ECC (Private Key): E<br>- QASM_SCA_COM (Private Key): E |
| EDDSA Signature generation | Generate an EDDSA digital signature | HSM_RET_OK | Private Key, Message, Hash Algorithm | Digital Signature | Digital Signature generation with EDDSA | Owner<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>Access User<br>- EDDSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E |
| LMS Signature generation | Generate an LMS digital signature | HSM_RET_OK | Private Key, Message, Hash Algorithm | Digital Signature | Digital Signature generation with LMS | Owner<br>- LMS Digital Signature Private Keys (Generated): E<br>Security Officer<br>- LMS Digital Signature Private Keys (Generated): E<br>Access User<br>- LMS Digital Signature Private Keys (Generated): E |
| HSS Signature generation | Generate a HSS digital signature | HSM_RET_OK | Private Key, Message, Hash Algorithm | Digital Signature | Digital Signature generation with HSS | Owner<br>- HSS Digital Signature Private Keys (Generated): E<br>- QASM_IA_HSS |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Private Key): E<br>- QASM_AA_HSS (Private Key): E<br>Security Officer<br>- HSS Digital Signature Private Keys (Generated): E<br>- QASM_IA_HSS (Private Key): E<br>- QASM_AA_HSS (Private Key): E<br>Access User<br>- HSS Digital Signature Private Keys (Generated): E<br>- QASM_IA_HSS (Private Key): E<br>- QASM_AA_HSS (Private Key): E |
| ML-DSA Signature generation | Generate an ML-DSA digital | HSM_RET_OK | Private Key, Message | Digital Signature | Digital Signature | Owner<br>- ML-DSA Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| (External Interface, Pure) | signature using the external signature interface in pure mode (full message input) in hedged mode. | | | | generation with ML-DSA (External Interface, Pure, Hedged) | Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Access User<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E |
| ML-DSA Signature generation (External | Generate an ML-DSA digital signature | HSM_RET_OK | Private Key, Message or PHM, | Digital Signature | Digital Signature generatio | Owner<br>- ML-DSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Interface, Pre-Hash) | using the external signature interface in pre-hash mode in hedged mode. Accepts either the full message (digest computed internally) or an externally computed digest (PHM), hashing with SHA2-512. | | Hash Algorithm | | n with ML-DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM) | Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Access User<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E |
| ML-DSA Signature generation (External Mu) | Generate an ML-DSA digital signature from an | HSM_RET_OK | Private Key, External Mu | Digital Signature | Digital Signature generation with | Owner<br>- ML-DSA Digital Signature Private |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | externally computed message representative (mu) supplied to the internal signature interface, in non-deterministic (hedged) mode. | | | | ML-DSA (Internal Interface, Hedged, External Mu) | Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>Access User<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E |
| RSA Signature generation | Generate an RSA digital signature | HSM_RET_OK | Private Key, Message, Hash Algorithm | Digital Signature | Digital Signature generation with RSA | Owner<br>- RSA Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>Access User<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E |
| RSA Signature generation primitive | Generate an RSA digital signature | HSM_RET_OK | Private Key, message digest | Digital Signature | RSA Signature Primitive 2.0 | Owner<br>- RSA Digital Signature Private Keys (Generated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ): E<br>- RSA Digital Signature Private Keys (Established): E<br>Security Officer<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>Access User<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E |
| SLH-DSA Signature generation (External Interface, Pure) | Generate an SLH-DSA digital signature using the external signature interface in | HSM_RET_OK | Private Key, Message | Digital Signature | Digital Signature generation with SLH-DSA (External | Owner<br>- SLH-DSA Digital Signature Private Keys (Generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | pure mode (full message input) in hedged mode. | | | | Interface, Pure, Hedged) | - SLH-DSA Digital Signature Private Keys (Established): E Security Officer<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E Access User<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E |
| SLH-DSA Signature generation (External Interface, Pre-Hash) | Generate an SLH-DSA digital signature using the external signature interface in pre-hash | HSM_RET_OK | Private Key, Message or PHM, Hash Algorithm | Digital Signature | Digital Signature generation with SLH-DSA (External Interface, | Owner<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | mode in hedged mode. Accepts either the full message (digest computed internally) or an externally computed digest (PHM), hashing with SHA2-512. | | | | PreHash, Hedged, Message)<br>Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM) | Digital Signature Private Keys (Established): E<br>Security Officer<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E<br>Access User<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E |
| ECDSA Signature verification | Verify the signature on a message signed with ECDSA | HSM_RET_OK | Public Key, Signature, Message, Hash Algorithm | Pass or Fail | Digital Signature verification with ECDSA | Owner<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Established): E<br>- C4A_TA_HSS: E<br>- QASM_IA_ECC (Public Key): E<br>- C4A_RCA_COM: E<br>- C4A_SCA_COM: E<br>- C4A_RCA: E<br>- C4A_SCA: E<br>- C4A_TA_ECDSA: E<br>- QASM_SCA_COM (Public Key): E<br>Security Officer<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- C4A_TA_HSS: E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - QASM_IA_ECC (Public Key): E<br>- C4A_RCA_COM: E<br>- C4A_SCA_COM: E<br>- C4A_RCA: E<br>- C4A_SCA: E<br>- C4A_TA_ECDSA: E<br>- QASM_SCA_COM (Public Key): E<br>Access User<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- QASM_IA_ECC (Public Key): E<br>- C4A_RCA_COM: E<br>- |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | C4A_SCA_COM: E<br>-<br>C4A_RCA: E<br>-<br>C4A_SCA: E<br>-<br>QASM_SCA_COM (Public Key): E |
| EDDSA Signature verification | Verify the signature on a message signed with EDDSA | HSM_RET_OK | Public Key, Signature, Message, Hash Algorithm | Pass or Fail | Digital Signature verification with EDDSA | Owner<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>Access User<br>- EDDSA Digital Signature Public Keys (generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - EDDSA Digital Signature Public Keys (Established): E |
| LMS Signature verification | Verify the signature on a message signed with LMS | HSM_RET_OK | Public Key, Signature, Message, Hash Algorithm | Pass or Fail | Digital Signature verification with LMS | Owner<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>Security Officer<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>Access User<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| HSS Signature verification | Verify the signature on a message signed with HSS | HSM_RET_OK | Public Key, Signature, Message, Hash Algorithm | Pass or Fail | Digital Signature generation with HSS | Owner<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>- QASM_IA_HSS (Public Key): E<br>- QASM_AA_HSS (Public Key): E<br>Security Officer<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>- QASM_IA_HSS (Public Key): E<br>- QASM_AA_HSS (Public Key): E<br>Access |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | User<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>- QASM_IA_HSS (Public Key): E<br>- QASM_AA_HSS (Public Key): E |
| ML-DSA Signature verification (External Interface, Pure) | Verify the signature on a message signed with ML-DSA using the external signature interface in pure mode (full message input). | HSM_RET_OK | Public Key, Signature, Message | Pass or Fail | Digital Signature verification with ML-DSA (External Interface, Pure) | Owner<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E<br>Access User<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E |
| ML-DSA Signature verification (External Interface, Pre-Hash) | Verify the signature on a message signed with ML-DSA using the external signature interface in pre-hash mode. Accepts either the full message (digest computed internally) or an externally computed digest (PHM), hashing with SHA2-512. | HSM_RET_OK | Public Key, Signature, Message or PHM, Hash Algorithm | Pass or Fail | Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, PHM) | Owner<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>Access User<br>- ML-DSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E |
| RSA Signature verification | Verify the signature on a message signed with RSA | HSM_RET_OK | Public Key, Signature, Message, Hash Algorithm | Pass or Fail | Digital Signature verification with RSA | Owner<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>Access User<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E |
| SLH-DSA Signature verification (External Interface, Pure) | Verify the signature on a message signed with SLH-DSA using the external signature interface in pure mode (full message input). | HSM_RET_OK | Public Key, Signature, Message | Pass or Fail | Digital Signature verification with SLH-DSA (External Interface, Pure) | Owner<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>Access User<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E |
| SLH-DSA Signature verification (External | Verify the signature on a message signed with SLH-DSA | HSM_RET_OK | Public Key, Signature, Message or PHM, | Pass or Fail | Digital Signature verification with | Owner<br>- SLH-DSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Interface, Pre-Hash) | using the external signature interface in pre-hash mode. Accepts either the full message (digest computed internally) or an externally computed digest (PHM), hashing with SHA2-512. | | Hash Algorithm | | SLH-DSA (External Interface, PreHash, Message) Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) | (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>Access User<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E |
| Key transport (key encapsulation using RSA) | Perform key encapsulation using RSA | HSM_RET_OK | Target key to be encapsulated, RSA public key | Encapsulated key | Key encapsulation with RSA (KTS-Encapsulation) | Owner<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | d): E<br>Security Officer<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Established): E<br>Access User<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Established): E |
| Key transport (key decapsulation using RSA) | Perform key decapsulation using RSA | HSM_RET_OK | Encapsulated key, UUID/reference to RSA private key | UUID/reference to decapsulated key | Key decapsulation with RSA (KTS-Decapsulation) | Owner<br>- RSA Key Decapsulation Private Key (Generated): E<br>- RSA Key Decapsulation Private Key (Established): E<br>Security Officer<br>- RSA Key Decapsulation Private Key |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E<br>- RSA Key Decapsulation Private Key (Established): E<br>Access User<br>- RSA Key Decapsulation Private Key (Generated): E<br>- RSA Key Decapsulation Private Key (Established): E |
| Key wrap with ML-KEM-AES | Key wrapping using AES-KW/KWP with key wrapping key obtained using ML-KEM | HSM_RET_OK | Target key to be wrapped, ML-KEM public key | wrapped target key, ML-KEM ciphertext | Key encapsulation with ML-KEM Key wrapping with AES-KW or AES-KWP | Owner<br>- ML-KEM Key Encapsulation Public Key (Generated): E<br>- ML-KEM Key Encapsulation Public Key (Established): E<br>- ML-KEM Shared Secret: G,E<br>Security Officer<br>- ML-KEM Key Encapsulation Public Key (Generated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ): E<br>- ML-KEM Key Encapsulation Public Key (Established): E<br>- ML-KEM Shared Secret: G,E Access User<br>- ML-KEM Key Encapsulation Public Key (Generated): E<br>- ML-KEM Key Encapsulation Public Key (Established): E<br>- ML-KEM Shared Secret: G,E |
| Key unwrap with ML-KEM-AES | Key unwrapping using AES-KW/KWP with key wrapping key obtained using ML-KEM | HSM_RET_OK | wrapped target key, UUID/reference to ML-KEM private key, ML-KEM ciphertext | UUID/reference to unwrapped key | Key decapsulation with ML-KEM Key unwrapping with AES-KW or AES-KWP | Owner<br>- ML-KEM Key Decapsulation Private Key (Generated): E<br>- ML-KEM Key Decapsulation Private Key (Established): E<br>- ML-KEM Shared Secret: G,E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Security Officer<br>- ML-KEM Key Decapsulation Private Key (Generated): E<br>- ML-KEM Key Decapsulation Private Key (Established): E<br>- ML-KEM Shared Secret: G,E<br>Access User<br>- ML-KEM Key Decapsulation Private Key (Generated): E<br>- ML-KEM Key Decapsulation Private Key (Established): E<br>- ML-KEM Shared Secret: G,E |
| Key wrap with ECDH-AES | Key wrapping using AES-KW/KWP with key wrapping key established using ECDHE | HSM_RET_OK | Target key to be wrapped, ECDH public key | Public ephemeral key concatenated with AES-KW/AES-KWP ciphertext | Key pair generation with ECDSA Key wrapping with AES-KW or AES-KWP | Owner<br>- ECDH Provider Private Key: G,E<br>- ECDH Recipient Public Key: R<br>- ECC |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Elliptic curve Diffie-Hellman key agreement | Shared Secret: E<br>- SP800-56Cr2 Derived Key: E<br>Security Officer<br>- ECDH Provider Private Key: G,E<br>- ECDH Recipient Public Key: R<br>- ECC Shared Secret: E<br>- SP800-56Cr2 Derived Key: E<br>Access User<br>- ECDH Provider Private Key: G,E<br>- ECDH Recipient Public Key: R<br>- ECC Shared Secret: E<br>- SP800-56Cr2 Derived Key: E |
| Key unwrap with ECDH-AES | Key unwrapping using AES-KW/KWP with key wrapping key established | HSM_RET_OK | Public ephemeral key, AES-KW/P ciphertext, UUID/reference to | UUID/reference to unwrapped key | Key unwrapping with AES-KW or AES-KWP Elliptic curve | Owner<br>- QASM_KE_ECC (Private Key): E<br>- ECDH Recipient |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | using ECDHE | | ECDH private key | | Diffie-Hellman key agreement | Public Key: R<br>- ECC Shared Secret: E<br>- SP800-56Cr2 Derived Key: E<br>Security Officer<br>- QASM_KE_ECC (Private Key): E<br>- ECDH Recipient Public Key: R<br>- ECC Shared Secret: E<br>- SP800-56Cr2 Derived Key: E<br>Access User<br>- QASM_KE_ECC (Private Key): E<br>- ECDH Recipient Public Key: R<br>- ECC Shared Secret: E<br>- SP800-56Cr2 Derived Key: E |
| Key wrap with AES- | Wrap a key using AES Key Wrap or | HSM_RET_OK | Target key to be wrapped, | Wrapped key | Key wrapping with | Owner<br>- AES-KW Symmetric |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| KW or AES-KWP | AES Key Wrap with Padding | | UUID/reference to key encryption key | | AES-KW or AES-KWP | Keys (Generated): E<br>- AES-KW Symmetric Keys (Established): E Security Officer<br>- AES-KW Symmetric Keys (Generated): E<br>- AES-KW Symmetric Keys (Established): E Access User<br>- AES-KW Symmetric Keys (Generated): E<br>- AES-KW Symmetric Keys (Established): E |
| Key unwrap with AES-KW or AES-KWP | Unwrap a key using AES Key Wrap or AES Key Wrap with Padding | HSM_RET_OK | Wrapped key, UUID/reference to key encryption key | UUID/reference to unwrapped key | Key unwrapping with AES-KW or AES-KWP | Owner<br>- AES-KW Symmetric Keys (Generated): E<br>- AES-KW Symmetric Keys (Established): E Security Officer<br>- AES-KW Symmetric Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E<br>- AES-KW Symmetric Keys (Established): E<br>Access User<br>- AES-KW Symmetric Keys (Generated): E<br>- AES-KW Symmetric Keys (Established): E |
| Export key (MofN) | Export a key from the module using Shamir Secret Sharing | HSM_RET_OK | Custodian's public keys, key wrap mechanism choice | Shards, shard keys, packed Backup Key Encryption Key (BKEK) | Key wrapping with AES-KW or AES-KWP<br>Key encapsulation with RSA (KTS-Encapsulation)<br>Key encapsulation with ML-KEM | Owner<br>- AES-KW Symmetric Keys (Generated): E,R<br>- AES-KW Symmetric Keys (Established): E,R<br>Security Officer<br>- AES-KW Symmetric Keys (Generated): E,R<br>- AES-KW Symmetric Keys (Established): E,R<br>Access User<br>- AES-KW Symmetric Keys (Generated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ): E,R<br>- AES-KW Symmetric Keys (Established): E,R |
| Import key (MofN) | Import a key into the module using Shamir Secret Sharing | HSM_RET_OK | Exported shards, exported shard keys, custodian's private keys, packed BKEK | N/A | Key unwrapping with AES-KW or AES-KWP<br>Key encapsulation with RSA (KTS-Encapsulation)<br>Key encapsulation with ML-KEM | Owner<br>- AES-KW Symmetric Keys (Generated): E,R<br>- AES-KW Symmetric Keys (Established): E,R<br>Security Officer<br>- AES-KW Symmetric Keys (Generated): E,R<br>- AES-KW Symmetric Keys (Established): E,R<br>Access User<br>- AES-KW Symmetric Keys (Generated): E,R<br>- AES-KW Symmetric Keys (Established): E,R |
| Change time policy | Change the mode of the time policy which dictates how the module tracks time | HSM_RET_OK | N/A | N/A | Digital Signature generation with ECDSA Digital | Owner<br>- ECDSA Digital Signature Private Keys (Generated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | (e.g. either accept external time or only track internal clock set at manufacturing) | | | | Signature generation with EDDSA Digital Signature generation with RSA Digital Signature generation with HSS Digital Signature generation with LMS Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged) Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message) Digital | ): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>- HSS Digital Signature Private Keys (Generated): E<br>- LMS Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM) | (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E |
| Transfer ownership | Transfer ownership of the module to a new owner | HSM_RET_OK | Signed authorization request, New Owner public key | N/A | Digital Signature verification with ECDSA | Owner<br>- Owner Public Key(C4A): E<br>- New Owner Public Key: W |
| Firmware update | Update the module firmware. Perform a firmware load test before loading new firmware. | HSM_RET_OK | New Firmware Image, Signed Firmware Image Manifest | N/A | Digital Signature verification with ECDSA Digital Signature verification with HSS Authenticated symmetric encryption with AES-GCM | Owner<br>- Owner Public Key(C4A): E<br>- New Owner Public Key: E<br>- C4A_TA_HSS: E<br>- C4A_TA_ECDSA: E<br>- AES-GCM Symmetric Keys (Generated): E<br>- AES-GCM Symmetric Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E<br>- TKA Wrapping Key for ATP0 (TKAWK0): E<br>- Tamper Key for ATP0 (TKA0): E<br>- TKA Wrapping Key for ATP1 (TKAWK1): E<br>- Tamper Key for ATP1 (TKA1): E<br>- Tamper Key for HSM (TKH): E<br>- Tamper Key for HaaP (TKU): E<br>- Zeroization Key for HaaP (ZKU): E |
| Add SO | Add a new Security Officer (SO) to the module | HSM_RET_OK | Security Officer Keypair | N/A | Digital Signature verification with ECDSA | Owner<br>- Security Officer Public Key: W |
| Remove SO | Remove a Security Officer (SO) from the module | HSM_RET_OK | N/A | N/A | None | Owner |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Add AU | Add Access User (AU) to the module | HSM_RET_OK | Access User Keypair | N/A | None | Owner Security Officer |
| Remove AU | Remove Access User (AU) from the module | HSM_RET_OK | N/A | N/A | None | Owner Security Officer |
| Login | Login to a role on the module | HSM_RET_OK | Owner, Security Officer or Access User's signed request | N/A | Digital Signature verification with ECDSA<br>Digital Signature verification with EDDSA<br>Digital Signature verification with LMS<br>Digital Signature verification with ML-DSA (External Interface, Pure)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Digital Signature | Owner<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | verification with ML-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with RSA<br>Digital Signature verification with SLH-DSA (External Interface, Pure)<br>Digital Signature verification with SLH-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM)<br>Digital Signatur | (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Security Officer<br>- ECDSA Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | e verification with HSS | Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Access User<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E |
| Logout | Logout of a role on the module | HSM_RET_OK | Owner, Security Officer or Access User's signed request | N/A | Digital Signature verification with ECDSA<br>Digital Signature verification with EDDSA<br>Digital Signature verification with LMS<br>Digital Signature verification with ML-DSA | Owner<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | (External Interface, Pure)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with RSA<br>Digital Signature verification with SLH-DSA (External Interface, Pure)<br>Digital Signature verification with SLH-DSA (External | d): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Interface, PreHash, Message) Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) Digital Signature verification with HSS | d): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Security Officer<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Established): E<br>Access User<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Public Keys (Established): E<br>- RSA<br>Digital Signature Public Keys (Generated): E<br>- RSA<br>Digital Signature Public Keys (Established): E<br>- SLH-DSA<br>Digital Signature Public Keys (Generated): E<br>- SLH-DSA<br>Digital Signature Public Keys (Established): E<br>- HSS<br>Digital Signature Public Keys (Generated): E<br>- HSS<br>Digital Signature Public Keys (Established): E |
| Show status / show version | Display a readout of the module's status | HSM_RET_OK | N/A | Status output | None | Owner<br>Security Officer<br>Access User<br>Unauthenticated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Self-test | Perform on-demand self-testing | HSM_RET_OK | N/A | Self-test results | Symmetric encryption with AES<br>Symmetric decryption with AES<br>Authenticated symmetric encryption with AES-GCM<br>Authenticated symmetric decryption with AES-GCM<br>Digital Signature generation with ECDSA<br>Digital Signature verification with ECDSA<br>Digital Signature generation with EDDSA<br>Digital Signature verificati | Owner<br>Security Officer<br>Access User<br>Unauthenticated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | on with EDDSA Random number generation with Hash DRBG Message authentication code with HMAC Key encapsulation with RSA (KTS-Encapsulation) Key decapsulation with RSA (KTS-Decapsulation) Digital Signature generation with LMS Digital Signature verification with LMS Digital Signature generation with ML-DSA (External Interface, | |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Pure, Hedged)<br>Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, Message)<br>Digital Signature verification with ML-DSA (External Interface, Pure)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Key encapsulation with ML-KEM<br>Key decapsulation with ML-KEM<br>Digital Signature generatio | |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | n with RSA<br>Digital Signature verification with RSA<br>Compute message digest with SHA<br>Extendable output functions<br>Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged)<br>Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message)<br>Digital Signature verification with SLH-DSA (External | |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Interface, Pure) Digital Signature verification with SLH-DSA (External Interface, PreHash, Message) Digital Signature generation with HSS Digital Signature verification with HSS | |
| Store data object | Store a data object in module memory | HSM_RET_OK | N/A | N/A | None | Owner Security Officer Access User Unauthenticated |
| Read data object | Read a data object from module memory | HSM_RET_OK | N/A | N/A | None | Owner Security Officer Access User Unauthenticated |
| Create user | Create a new Access User, or Security Officer | HSM_RET_OK | Owner's or Security Officer's signed authorization request, | N/A | Digital Signature verification with ECDSA Digital | Owner - ECDSA Digital Signature Public Keys (Established): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | New User public key | | Signature verification with EDDSA<br>Digital Signature verification with LMS<br>Digital Signature verification with ML-DSA (External Interface, Pure)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with RSA<br>Digital | - RSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>Security Officer<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Signature verification with SLH-DSA (External Interface, Pure)<br>Digital Signature verification with SLH-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with HSS | (Established): W,E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): W,E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): W,E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): W,E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): W,E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): W,E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): W,E<br>Access User<br>- ECDSA Digital Signature Public Keys (Established): W<br>- RSA Digital Signature Public Keys (Established): W<br>- EDDSA Digital Signature Public Keys (Established): W<br>- HSS Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Signature Public Keys (Established): W<br>- ML-DSA Digital Signature Public Keys (Established): W<br>- LMS Digital Signature Public Keys (Established): W<br>- SLH-DSA Digital Signature Public Keys (Established): W |
| Create quorum | Create a quorum in the object model connected to a set of authorities | HSM_RET_OK | N/A | UUID of created quorum | None | Owner<br>Security Officer Access User |
| Extraction under authorization | Performs the extraction of a target key only after exceeding the threshold of signatures required from an associated quorum of users. | HSM_RET_OK | N/A | Packed or wrapped key | Digital Signature generation with ECDSA<br>Digital Signature generation with EDDSA<br>Key encapsulation with RSA (KTS-Encapsulation) | Owner<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Digital Signature generation with LMS<br>Digital Signature generation with ML-DSA (External Interface, Pure, Hedged)<br>Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, Message)<br>Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM)<br>Key encapsulation with ML-KEM<br>Digital Signature generatio | - EDDSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | n with RSA<br>Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged)<br>Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message)<br>Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM)<br>Digital Signature generation with HSS | - ML-DSA Digital Signature Public Keys (Established): E<br>- ML-KEM Key Encapsulation Public Key (Generated): E<br>- ML-KEM Key Encapsulation Public Key (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Security Officer<br>- ECDSA Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Established): E<br>- LMS Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- ML-KEM Key Encapsulation Public Key (Generated): E<br>- ML-KEM Key Encapsulation Public Key (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Access User<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E<br>- RSA Key Encapsulation Public Key (Generated): E<br>- RSA Key Encapsulation Public Key (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Established): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- ML-KEM Key Encapsulation Public Key (Generated): E<br>- ML-KEM Key Encapsulati |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | on Public Key (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E |
| Object packing/unpacking | Perform packing and unpacking of a target key | HSM_RET_OK | Unpacked/ packed key | Packed/unpacked key | Key wrapping with AES-KW or AES-KWP<br>Key unwrapping with AES-KW or AES-KWP<br>Key verification with ECDSA<br>Key verification with EDDSA | Owner<br>- RSA Key Decapsulation Private Key (Generated): E,R,W<br>- RSA Key Decapsulation Private Key (Established): E,R,W<br>- RSA Key Encapsulation Public Key (Generated): E,R,W<br>- RSA Key |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Key encapsulation with RSA (KTS-Encapsulation)<br>Key decapsulation with RSA (KTS-Decapsulation)<br>Key encapsulation with ML-KEM<br>Key decapsulation with ML-KEM<br>Compute message digest with SHA | Encapsulation Public Key (Established): E,R,W<br>- ML-KEM<br>Key Decapsulation Private Key (Generated): E,R,W<br>- ML-KEM<br>Key Decapsulation Private Key (Established): E,R,W<br>- ML-KEM<br>Key Encapsulation Public Key (Generated): E,R,W<br>- ML-KEM<br>Key Encapsulation Public Key (Established): E,R,W<br>- AES-KW<br>Symmetric Keys (Generated): E,R,W<br>- AES-KW<br>Symmetric Keys (Established): E,R,W<br>- AES<br>Symmetric Keys (Generated): R,W |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - AES Symmetric Keys (Established): R,W<br>- Owner Public Key(C4A): E<br>- Security Officer Public Key: E<br>- AES-GCM Symmetric Keys (Generated): E,R<br>- AES-GCM Symmetric Keys (Established): E,R<br>- RSA Digital Signature Private Keys (Generated): W,R<br>- RSA Digital Signature Private Keys (Established): W,R<br>- RSA Digital Signature Public Keys (Generated): W,R<br>- RSA Digital Signature Public Keys (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | d): W,R<br>- ECDSA Digital Signature Private Keys (Generated): W,R<br>- ECDSA Digital Signature Private Keys (Established): W,R<br>- ECDSA Digital Signature Public Keys (Generated): W,R<br>- ECDSA Digital Signature Public Keys (Established): W,R<br>- EDDSA Digital Signature Private Keys (Generated): W,R<br>- EDDSA Digital Signature Private Keys (Established): W,R<br>- EDDSA Digital Signature Public Keys (generated): W,R<br>- EDDSA |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Public Keys (Established): W,R<br>- LMS Digital Signature Public Keys (Generated): W,R<br>- LMS Digital Signature Public Keys (Established): W,R<br>- ML-DSA Digital Signature Private Keys (Generated): W,R<br>- ML-DSA Digital Signature Private Keys (Established): W,R<br>- ML-DSA Digital Signature Public Keys (Generated): W,R<br>- ML-DSA Digital Signature Public Keys (Established): W,R<br>- SLH-DSA Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): W,R<br>- SLH-DSA Digital Signature Private Keys (Established): W,R<br>- SLH-DSA Digital Signature Public Keys (Generated): W,R<br>- SLH-DSA Digital Signature Public Keys (Established): W,R<br>- HSS Digital Signature Public Keys (Generated): W,R<br>- HSS Digital Signature Public Keys (Established): W,R<br>- KBKDF Derived Key: W,R<br>- ECDH Derived Key: W,R<br>Security Officer<br>- RSA Key Decapsulation Private Key (Generated): E,R,W<br>- RSA Key |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Decapsulation Private Key (Established): E,R,W<br>- RSA Key Encapsulation Public Key (Generated): E,R,W<br>- RSA Key Encapsulation Public Key (Established): E,R,W<br>- ML-KEM Key Decapsulation Private Key (Generated): E,R,W<br>- ML-KEM Key Decapsulation Private Key (Established): E,R,W<br>- ML-KEM Key Encapsulation Public Key (Generated): E,R,W<br>- ML-KEM Key Encapsulation Public Key (Established): E,R,W<br>- AES-KW Symmetric Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E,R,W<br>- AES-KW Symmetric Keys (Established): E,R,W<br>- AES Symmetric Keys (Generated): R,W<br>- AES Symmetric Keys (Established): R,W<br>- Owner Public Key(C4A): W<br>- Security Officer Public Key: W<br>- AES-GCM Symmetric Keys (Generated): W,R<br>- AES-GCM Symmetric Keys (Established): W,R<br>- RSA Digital Signature Private Keys (Generated): W,R<br>- RSA Digital Signature Private Keys (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | d): W,R<br>- RSA Digital Signature Public Keys (Generated): W,R<br>- RSA Digital Signature Public Keys (Established): W,R<br>- ECDSA Digital Signature Private Keys (Generated): W,R<br>- ECDSA Digital Signature Private Keys (Established): W,R<br>- ECDSA Digital Signature Public Keys (Generated): W,R<br>- ECDSA Digital Signature Public Keys (Established): W,R<br>- EDDSA Digital Signature Private Keys (Generated): W,R<br>- EDDSA Digital |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Signature Private Keys (Established): W,R<br>- EDDSA Digital Signature Public Keys (generated): W,R<br>- EDDSA Digital Signature Public Keys (Established): W,R<br>- LMS Digital Signature Public Keys (Generated): W,R<br>- LMS Digital Signature Public Keys (Established): W,R<br>- ML-DSA Digital Signature Private Keys (Generated): W,R<br>- ML-DSA Digital Signature Private Keys (Established): W,R<br>- ML-DSA Digital Signature Public Keys (Generated |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ): W,R<br>- ML-DSA Digital Signature Public Keys (Established): W,R<br>- SLH-DSA Digital Signature Private Keys (Generated): W,R<br>- SLH-DSA Digital Signature Private Keys (Established): W,R<br>- SLH-DSA Digital Signature Public Keys (Generated): W,R<br>- SLH-DSA Digital Signature Public Keys (Established): W,R<br>- HSS Digital Signature Public Keys (Generated): W,R<br>- HSS Digital Signature Public Keys (Established): W,R<br>- KBKDF Derived Key: W,R |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - ECDH Derived Key: W,R Access User<br>- RSA Key Decapsulation Private Key (Generated): E,R,W<br>- RSA Key Decapsulation Private Key (Established): E,R,W<br>- RSA Key Encapsulation Public Key (Generated): E,R,W<br>- RSA Key Encapsulation Public Key (Established): E,R,W<br>- ML-KEM Key Decapsulation Private Key (Generated): E,R,W<br>- ML-KEM Key Decapsulation Private Key (Established): E,R,W<br>- ML-KEM Key Encapsulation Public Key |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Generated): E,R,W<br>- ML-KEM Key Encapsulation Public Key (Established): E,R,W<br>- AES-KW Symmetric Keys (Generated): E,R,W<br>- AES-KW Symmetric Keys (Established): E,R,W<br>- AES Symmetric Keys (Generated): R,W<br>- AES Symmetric Keys (Established): R,W<br>- Owner Public Key(C4A): W<br>- Security Officer Public Key: W<br>- AES-GCM Symmetric Keys (Generated): W,R<br>- AES-GCM Symmetric Keys (Established): W,R<br>- RSA |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Private Keys (Generated): W,R<br>- RSA Digital Signature Private Keys (Established): W,R<br>- RSA Digital Signature Public Keys (Generated): W,R<br>- RSA Digital Signature Public Keys (Established): W,R<br>- ECDSA Digital Signature Private Keys (Generated): W,R<br>- ECDSA Digital Signature Private Keys (Established): W,R<br>- ECDSA Digital Signature Public Keys (Generated): W,R<br>- ECDSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Established): W,R<br>- EDDSA Digital Signature Private Keys (Generated): W,R<br>- EDDSA Digital Signature Private Keys (Established): W,R<br>- EDDSA Digital Signature Public Keys (generated): W,R<br>- EDDSA Digital Signature Public Keys (Established): W,R<br>- LMS Digital Signature Public Keys (Generated): W,R<br>- LMS Digital Signature Public Keys (Established): W,R<br>- ML-DSA Digital Signature Private Keys (Generated): W,R |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - ML-DSA Digital Signature Private Keys (Established): W,R<br>- ML-DSA Digital Signature Public Keys (Generated): W,R<br>- ML-DSA Digital Signature Public Keys (Established): W,R<br>- SLH-DSA Digital Signature Private Keys (Generated): W,R<br>- SLH-DSA Digital Signature Private Keys (Established): W,R<br>- SLH-DSA Digital Signature Public Keys (Generated): W,R<br>- SLH-DSA Digital Signature Public Keys (Established): W,R<br>- HSS Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Public Keys (Generated): W,R<br>- HSS Digital Signature Public Keys (Established): W,R<br>- KBKDF Derived Key: W,R<br>- ECDH Derived Key: W,R |
| Attestation | Create an attestation signature | HSM_RET_OK | N/A | N/A | Digital Signature generation with ECDSA<br>Digital Signature generation with EDDSA<br>Digital Signature generation with LMS<br>Digital Signature generation with ML-DSA (External Interface, Pure, Hedged)<br>Digital Signature generation with | Owner<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>- LMS Digital Signature Private |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | ML-DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM) Digital Signature generation with RSA Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged) Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message) | Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E<br>- HSS Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM)<br>Digital Signature generation with HSS | Private Keys (Generated): E<br>Security Officer<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>- LMS Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Generated): E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - ML-DSA Digital Signature Private Keys (Established): E<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E<br>- HSS Digital Signature Private Keys (Generated): E<br>Access User<br>- ECDSA Digital Signature Private |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>- LMS Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Generated): E<br>- ML-DSA Digital Signature Private Keys (Established): E<br>- RSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Private Keys (Generated): E<br>- RSA Digital Signature Private Keys (Established): E<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E<br>- HSS Digital Signature Private Keys (Generated): E |
| Pre-authorized operations | An externally submitted request is received and performed if enough members of a relevant quorum sign the request. | HSM_RET_OK | Submitted request | N/A | Digital Signature verification with ECDSA Digital Signature verification with EDDSA Digital Signature verificati | Owner<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | on with LMS<br>Digital Signature verification with ML-DSA (External Interface, Pure)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with ML-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with RSA<br>Digital Signature verification with SLH-DSA (External Interface, Pure) | Public Keys (generated): E<br>- EDDSA Digital Signature<br>Public Keys (Established): E<br>- LMS Digital Signature<br>Public Keys (Generated): E<br>- LMS Digital Signature<br>Public Keys (Established): E<br>- ML-DSA Digital Signature<br>Public Keys (Generated): E<br>- ML-DSA Digital Signature<br>Public Keys (Established): E<br>- RSA Digital Signature<br>Public Keys (Generated): E<br>- RSA Digital Signature<br>Public Keys (Established): E<br>- SLH-DSA Digital Signature |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Digital Signature verification with SLH-DSA (External Interface, PreHash, Message)<br>Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM)<br>Digital Signature verification with HSS | Public Keys (Generated): E<br>- SLH-DSA Digital Signature<br>Public Keys (Established): E<br>- HSS Digital Signature<br>Public Keys (Generated): E<br>- HSS Digital Signature<br>Public Keys (Established): E<br>Security Officer<br>- ECDSA Digital Signature<br>Public Keys (Generated): E<br>- ECDSA Digital Signature<br>Public Keys (Established): E<br>- EDDSA Digital Signature<br>Public Keys (generated): E<br>- EDDSA Digital Signature<br>Public Keys (Established): E<br>- LMS |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Public Keys (Generated): E<br>- LMS<br>Digital Signature Public Keys (Established): E<br>- ML-DSA<br>Digital Signature Public Keys (Generated): E<br>- ML-DSA<br>Digital Signature Public Keys (Established): E<br>- RSA<br>Digital Signature Public Keys (Generated): E<br>- RSA<br>Digital Signature Public Keys (Established): E<br>- SLH-DSA<br>Digital Signature Public Keys (Generated): E<br>- SLH-DSA<br>Digital Signature Public Keys (Established): E<br>- HSS |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys (Established): E<br>Access User<br>- ECDSA Digital Signature Public Keys (Generated): E<br>- ECDSA Digital Signature Public Keys (Established): E<br>- EDDSA Digital Signature Public Keys (generated): E<br>- EDDSA Digital Signature Public Keys (Established): E<br>- LMS Digital Signature Public Keys (Generated): E<br>- LMS Digital Signature Public Keys (Establishe |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | d): E<br>- ML-DSA Digital Signature Public Keys (Generated): E<br>- ML-DSA Digital Signature Public Keys (Established): E<br>- RSA Digital Signature Public Keys (Generated): E<br>- RSA Digital Signature Public Keys (Established): E<br>- SLH-DSA Digital Signature Public Keys (Generated): E<br>- SLH-DSA Digital Signature Public Keys (Established): E<br>- HSS Digital Signature Public Keys (Generated): E<br>- HSS Digital Signature Public Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | (Established): E |
| Zeroize module | Erasure of the Zeroization Key For HaaP (ZKU), which leaves behind all other keys in an encrypted state. | N/A | N/A | N/A | Digital Signature verification with ECDSA<br>Digital Signature verification with EDDSA<br>Digital Signature verification with RSA<br>Digital Signature verification with HSS<br>Digital Signature verification with LMS<br>Digital Signature verification with SLH-DSA (External Interface, Pure)<br>Digital Signature verification with SLH- | Owner - Zeroization Key for HaaP (ZKU): Z<br>- ECDSA Digital Signature Private Keys (Generated): E<br>- ECDSA Digital Signature Private Keys (Established): E<br>- EDDSA Digital Signature Private Keys (Generated): E<br>- EDDSA Digital Signature Private Keys (Established): E<br>- RSA Digital Signature Private Keys (Generated): E<br>- RSA Digital Signature Private Keys |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | DSA (External Interface, PreHash, Message) Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) | (Established): E<br>- HSS Digital Signature Private Keys (Generated): E<br>- LMS Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Generated): E<br>- SLH-DSA Digital Signature Private Keys (Established): E |

Table 13: Approved Services

## 4.4 Non-Approved Services

N/A for this module.

## 4.5 External Software/Firmware Loaded

The module supports loading a part of firmware which is upgradable i.e., OP.BIN. Firmware loading is accomplished using the “Firmware update” service, which can be initiated by a Crypto Officer during normal operation. The firmware load test performs approved authentication by verifying the ECDSA and HSS signatures. The module requires any Firmware Update service to be authenticated by the Crypto Officer before processing the request. All data output is inhibited during the execution of the firmware load test and the firmware loading process. Any firmware loaded into this module that is not shown on the module certificate is out of the scope of this validation and requires a separate FIPS 140-3 validation

# 5 Software/Firmware Security

## 5.1 Integrity Techniques

The module implements an approved integrity technique in two parts. First the module verifies the CRC-32 of the first stage bootloader, i.e., FSBL. The second part consists of verifying the AES-GMAC authentication tag of the OP.BIN when the firmware is decrypted. The OP.BIN firmware component is stored in the eMMC, located within the cryptographic boundary, and can be upgraded through the “Firmware update” service.

## 5.2 Initiate on Demand

Initiation of the integrity test on demand can be accomplished by rebooting the module.

# 6 Operational Environment

## 6.1 Operational Environment Type and Requirements

**Type of Operational Environment**: Limited

# 7 Physical Security

## 7.1 Mechanisms and Actions Required

| Mechanism | Inspection Frequency | Inspection Guidance |
|---|---|---|
| Tamper Label | N/A | As needed, the Crypto Officer shall inspect the module for any signs of physical tampering by observing whether tamper labels have been removed or manipulated. In the event of a confirmed tamper label breach, the module should be returned to Crypto4A for service. |

Table 14: Mechanisms and Actions Required

## 7.2 User Placed Tamper Seals

**Number:** 2

**Placement:** N/A

**Surface Preparation:** Not applicable because tamper seals are applied to the module before shipping.

**Operator Responsible for Securing Unused Seals:** N/A

**Part Numbers:** N/A

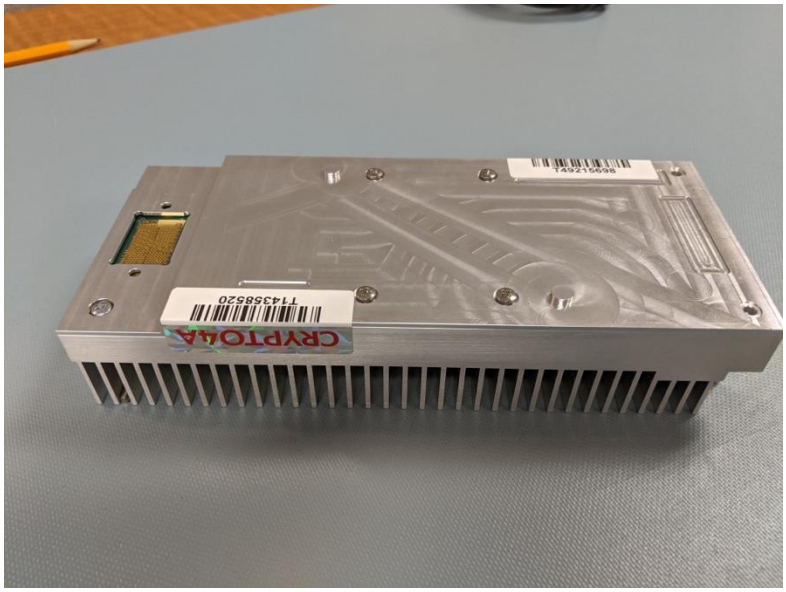



Figure 4: both tamper seals applied to module

## 7.4 Hardness Testing Temperature Ranges

| Temperature Type | Temperature |
|---|---|
| LowTemperature | -10C |
| HighTemperature | 70C |

Table 15: Hardness Testing Temperatures

## 7.5 EFP/EFT Information

| Temp/Voltage Type | Temperature or Voltage | EFP or EFT | Result |
|---|---|---|---|
| LowTemperature | -10C | EFP | Shutdown |
| HighTemperature | 70C | EFP | Shutdown |
| LowVoltage | 10.8V(VDD), 2.5V(ATP) | EFP | Shutdown |
| HighVoltage | 14V(VDD), 5.8V(ATP) | EFP | Shutdown |

Table 16: EFP/EFT Information

# 8 Non-Invasive Security

Not applicable to this module.

# 9 Sensitive Security Parameters Management

## 9.1 Storage Areas

| Storage Area Name | Description | Persistence Type |
|---|---|---|
| eMMC Flash | Stores encrypted firmware and encrypted user keys and certificates. | Static |
| FPGA eFuses | Objects such as the PPK hash are burned into the eFuses as permanently recorded, write-restricted data. | Static |
| ATP0 FRAM | FRAM on Anti-Tamper Processor 0 | Static |
| ATP1 FRAM | FRAM on Anti-Tamper Processor 1 | Static |
| ARTHUR FRAM | FRAM on the ARTHUR chip | Static |
| ARTHUR DRAM | DRAM on the ARTHUR chip | Dynamic |
| ATP0 SRAM | SRAM on Anti-Tamper Processor 0 that is constantly cycling TKA0 | Dynamic |
| ATP1 SRAM | SRAM on Anti-Tamper Processor 1 that is constantly cycling TKA1 | Dynamic |

Table 17: Storage Areas

## 9.2 SSP Input-Output Methods

| Name | From | To | Format Type | Distribution Type | Entry Type | SFI or Algorithm |
|---|---|---|---|---|---|---|
| Unwrap with AES | Outside | eMMC Flash | Encrypted | Automated | Electronic | Key unwrapping with AES-KW or AES-KWP |
| Unwrap with RSA | Outside | eMMC Flash | Encrypted | Automated | Electronic | Key decapsulation with RSA (KTS-Decapsulation) |
| MoFN (Import) | Outside | eMMC Flash | Encrypted | Automated | Electronic | Key unwrapping with AES-KW or AES-KWP |
| Unpack with AES | Outside | eMMC Flash | Encrypted | Automated | Electronic | Key unwrapping with AES-KW or AES-KWP |
| Wrap with AES | eMMC Flash | Outside | Encrypted | Automated | Electronic | Key wrapping with AES-KW or AES-KWP |
| Wrap with RSA | eMMC Flash | Outside | Encrypted | Automated | Electronic | Key encapsulation with RSA (KTS-Encapsulation) |
| MoFN (Export) | eMMC Flash | Outside | Encrypted | Automated | Electronic | Key wrapping with AES-KW or AES-KWP |

| Name | From | To | Format Type | Distribution Type | Entry Type | SFI or Algorithm |
|---|---|---|---|---|---|---|
| Pack with AES | eMMC Flash | Outside | Encrypted | Automated | Electronic | Key wrapping with AES-KW or AES-KWP |
| Import Recipient Public Key | Outside | ARTHUR DRAM | Plaintext | Automated | Electronic | Elliptic curve Diffie-Hellman key agreement |

Table 18: SSP Input-Output Methods

## 9.3 SSP Zeroization Methods

While the module is performing zeroization, services are not available, and data output (via the data output interface) is inhibited until the operation is successfully completed.

| Zeroization Method | Description | Rationale | Operator Initiation |
|---|---|---|---|
| Tamper Zeroization | When an intrusion is detected, the module explicitly zeroizes the TKU, TKH, TKA0 and TKA1. Once zeroization completes, the module sends a high signal through a ALERT0 pin on the AMI to indicate successful zeroization. Restarts QASM which also zeroizes all SSPs held in RAM. | The TKU, TKA0, TKA1 are root keys used to encrypt all child keys that encrypt every other SSP stored on the module. When either TKU, TKA0 or TKA1, keys are zeroized, every other key on the module is rendered irretrievable and only the tamper image can be loaded. TKH is used to decrypt the firmware. when it is zeroized, the module firmware will not boot. | When a tamper event occurs. |
| Zeroize module service | Pass a series of commands to explicitly zeroize all end user objects stored on the module. Once zeroization completes, the module sends a high signal through a ALERT0 pin on | The owner performs a procedure that explicitly zeroizes the ZKU which renders every end user object on the module irretrievable. This is equivalent to resetting the module back to factory condition. | Initiated by the module owner by submitting the following series of commands.1.PO_UUID=$(skm find owner class policy quick \| cut -d":" -f2)<br>2.skm request prepare target $PO_UUID zeroize out template.txt<br>3.skm key generate EC-P256 sign meta-data fips.function initiator |

| Zeroization Method | Description | Rationale | Operator Initiation |
|---|---|---|---|
| | the AMI to indicate successful zeroization. | | 4.PR_UUID=$(skm find meta-data fips.function initiator class private-key quick \| cut -d":" -f2)<br>5.PU_UUID=$(skm find meta-data fips.function initiator class public-key quick \| cut -d":" -f2)<br>6.skm sign $PR_UUID in template.txt out template.sig<br>7.AR_UUID=$(skm request create $PU_UUID signature template.sig template template.txt meta-data<br>8.fips.function request \| cut -d":" -f2)<br>9.skm request get $AR_UUID out request.txt |
| Manual Zeroization | Activate a physical interface on the module to explicitly zeroize all end user objects stored in the module. Once zeroization completes, the module sends a high signal through a ALERT0 pin on the AMI to indicate successful zeroization. | An operator with physical access to the module can hold the power button as well as a pinhole button in the front of the chassis for five seconds to explicitly zeroize the ZKU which renders every end user object in the IKS and GKS key stores irretrievable. This is equivalent to resetting the module back to factory condition. | Initiated by an operator with physical access to the module. |
| Automatic | Automatically implicitly zeroized by the module when no longer needed | Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable. | N/A |

Table 19: SSP Zeroization Methods

## 9.4 SSPs

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| TKA Wrapping Key for ATP0 (TKAWK0) | Used to decrypt TKA0. | 256 bits - 256 | Symmetric - CSP | | | Authenticated symmetric decryption with AES-GCM |
| Tamper Key for ATP0 (TKA0) | Key used to decrypt TKH, TKU and ZKU splits associated to ATP0. | 256 bits - 256 bits | Symmetric - CSP | | | Authenticated symmetric decryption with AES-GCM |
| TKA Wrapping Key for ATP1 (TKAWK1) | Used to decrypt TKA1. | 256 bits - 256 | Symmetric - CSP | | | Authenticated symmetric decryption with AES-GCM |
| Tamper Key for ATP1 (TKA1) | Key used to decrypt TKH, TKU and ZKU split associated to ATP1. | 256 bits - 256 bits | Symmetric - CSP | | | Authenticated symmetric decryption with AES-GCM |
| Tamper Key for HSM (TKH) | key used to decrypt module's firmware image. | 256 bits - 256 bits | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 Section 6.3 method 2) | | Authenticated symmetric decryption with AES-GCM |
| Tamper Key for HaaP (TKU) | The TKU is used to encrypt objects pertaining to the appliance that embeds the QASM | 256 bits - 256 bits | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 Section 6.3 method 2) | | Authenticated symmetric decryption with AES-GCM |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| Zeroization Key for HaaP (ZKU) | Key used to decrypt the TMK. Zeroized in the event of a tamper detected in the HaaP | 256 bits - 256 bits | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 Section 6.3 method 2) | | Authenticated symmetric decryption with AES-GCM |
| Tenant Master Key (TMK) | A key used to encrypt and decrypt all objects stored in the IKSs, GKS, OKS, and PKS | 256 bits - 256 | Symmetric - CSP | | | Authenticated symmetric encryption with AES-GCM Authenticated symmetric decryption with AES-GCM |
| DRBG Entropy Input | An entropy value used to instantiate the DRBG | 2048 bits - 256 | Entropy Source - CSP | ESV | | Random number generation with Hash DRBG |
| DRBG V | Part of the DRBG's secret state | 888 bits - 256 | DRBG - CSP | Hash DRBG (A5631) | | Random number generation with Hash DRBG |
| DRBG C | Part of the DRBG's secret state | 888 bits - 256 | DRBG - CSP | Hash DRBG (A5631) | | Random number generation with Hash DRBG |
| DRBG Seed | Random seed data obtained from the conditioned output of the entropy source | 888 bits - 256 | Entropy Source - CSP | Hash DRBG (A5631) | | Random number generation with Hash DRBG |
| C4A_TA_ECDSA | An ECDSA public key serving as a trust anchor | N/A - N/A | ECDSA public key - PSP | | | Digital Signature verification |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | used to validate new QASM FW images. | | | | | with ECDSA |
| C4A_TA_HSS | A HSS public key serving as a trust anchor used to validate new QASM FW images. | N/A - N/A | HSS public key - PSP | | | Digital Signature verification with HSS |
| C4A_RCA | Root certificate for Crypto4A. Used to validate IA (Identity & Authentication), KE (Key Exchange) and AA (Assertion Authority) sub-CA certificates | N/A - N/A | Certificate - PSP | | | Digital Signature verification with ECDSA |
| C4A_SCA | An Identity and Authentication sub-CA certificate used to validate IA functionality | N/A - N/A | Certificate - PSP | | | Digital Signature verification with ECDSA |
| QASM_IA_HSS (Private Key) | HSS private signing key (single root secret SEED I index per RFC 8554 Appendix A) for I&A (identification and | 416 bits - 256 | Private HSS - CSP | | | Digital Signature generation with HSS |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | authentication) | | | | | |
| QASM_IA_HSS (Public Key) | HSS public signature validation key for I&A | 480 bits - 256 | Public HSS - PSP | | | Digital Signature verification with HSS |
| QASM_IA_ECC (Private Key) | ECDSA private signing key (including intermediate values) for I&A | P-384 - 192 | Private ECDSA - CSP | | | Digital Signature generation with ECDSA |
| QASM_IA_ECC (Public Key) | ECDSA public signature validation key for I&A | P-384 - 192 | Public ECDSA - PSP | | | Digital Signature verification with ECDSA |
| QASM_KE_ECC (Private Key) | ECDH private key (including intermediate values) for performing key exchanges | P-384 - 192 | Private ECDH - CSP | | | Elliptic curve Diffie-Hellman key agreement |
| QASM_KE_ECC (Public Key) | ECDH public key for performing key exchanges | P-384 - 192 | Public ECDH - PSP | | | Elliptic curve Diffie-Hellman key agreement |
| QASM_AA_HSS (Private Key) | HSS private signing key (single root secret SEED I index per RFC 8554 Appendix A) used to sign attestation claims | 416 bits - 256 | Private HSS - CSP | | | Digital Signature generation with HSS |
| QASM_AA_HSS (Public Key) | HSS public key used to verify attestation | 480 bits - 256 | Public HSS - PSP | | | Digital Signature verification with HSS |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | claims from other QASMs | | | | | |
| QASM_AA_ECC (Private Key) | ECDH private key (including intermediate values) used to sign attestation claims | P-384 - 192 | Private ECDH - CSP | | | Elliptic curve Diffie-Hellman key agreement |
| QASM_AA_ECC (Public Key) | ECDH public key used to verify attestation claims from other QASMs | P-384 - 192 | Public ECDH - PSP | | | Elliptic curve Diffie-Hellman key agreement |
| C4A_RCA_COM | A root certificate used to validate the TLS certificate chains of the SBCs tied to the module | N/A - N/A | Certificate - PSP | | | Digital Signature verification with ECDSA |
| C4A_SCA_COM | A sub-CA used to issue (TLS) communication certificates to the module's SBCs | N/A - N/A | Certificate - PSP | | | Digital Signature verification with ECDSA |
| QASM_SCA_COM (Private Key) | Private key (including intermediate values) of the Sub CA for communication certificates | P-384 - 192 | ECDSA Private - CSP | | | Digital Signature generation with ECDSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| QASM_SCA_COM (Public Key) | Public key of the Sub CA for communication certificates | P-384 - 192 | ECDSA Public - PSP | | | Digital Signature verification with ECDSA |
| Owner Public Key(C4A) | A public key loaded at manufacturing and used to validate signatures on requests to modify the policy. ownership transfer replaces Owner Public Key(C4A) with an approved new public key of choice (e.g. ECDSA, HSS, LMS, ML-DSA, etc.) | P-384 - 192 | ECDSA Public - PSP | | | Digital Signature verification with ECDSA |
| New Owner Public Key | The incoming owner public key written to the module during ownership transfer. It replaces Owner Public Key(C4A) as the key used to validate signatures on requests | Dependent on selected algorithm (e.g. P-384 for ECDSA, ML-DSA-44/65/87 for ML-DSA) - 128-256 | Public - PSP | | | Digital Signature verification with ECDSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | to modify the policy. May be an approved public key of the owner's choice (e.g. ECDSA, HSS, LMS, ML-DSA). | | | | | |
| Security Officer Public Key | A public key used to authenticate the SO | P-384 - 192 | ECDSA Public - PSP | | | Digital Signature verification with ECDSA |
| AES Symmetric Keys (Generated) | Keys for performing unauthenticated AES encryption and decryption, generated by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 section 4 example 1, B = U) | | Symmetric encryption with AES Symmetric decryption with AES |
| AES Symmetric Keys (Established) | Keys for performing unauthenticated AES encryption and decryption, established by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Symmetric encryption with AES Symmetric decryption with AES |
| AES-KW Symmetric Keys (Generated) | Keys for performing AES-KW or AES-KWP key wrapping and unwrapping, generated by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 section 4 example 1, B = U) | | Key wrapping with AES-KW or AES-KWP Key unwrapping with AES-KW or AES-KWP |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| AES-KW Symmetric Keys (Established) | Keys for performing AES-KW or AES-KWP key wrapping and unwrapping, established by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key wrapping with AES-KW or AES-KWP Key unwrapping with AES-KW or AES-KWP |
| AES-GCM Symmetric Keys (Generated) | Keys for performing AES-GCM encryption and decryption, generated by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | Symmetric Key Generation (SP800-133rev2 section 4 example 1, B = U) | | Authenticated symmetric encryption with AES-GCM Authenticated symmetric decryption with AES-GCM |
| AES-GCM Symmetric Keys (Established) | Keys for performing AES-GCM encryption and decryption, established by the module | 128, 192 or 256 bits - 128, 192, 256 | Symmetric - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Authenticated symmetric encryption with AES-GCM Authenticated symmetric decryption with AES-GCM |
| HMAC Symmetric Keys (Established) | Keys used for generating message authentication codes with HMAC, established by the module | 112-8192 bits - 112-256 | Symmetric - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Message authentication code with HMAC |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| RSA Digital Signature Private Keys (Generated) | Keys (including intermediate values) for performing RSA digital signature generation, generated by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Private - CSP | Key pair generation with RSA | | Digital Signature generation with RSA |
| RSA Digital Signature Private Keys (Established) | Keys (including intermediate values) for performing RSA digital signature generation, established by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature generation with RSA |
| RSA Digital Signature Public Keys (Generated) | Keys for performing RSA digital signature verification, generated by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Public - PSP | Key pair generation with RSA | | Digital Signature verification with RSA |
| RSA Digital Signature Public Keys (Established) | Keys for performing RSA digital signature verification, established by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~150, ~200 | RSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with RSA |
| RSA Key Decapsulation Private Key (Generated) | Key (including intermediate values) for performing RSA key | 2048, 3072, 4096 or 8192 bits - 112, 128, | RSA Private - CSP | Key pair generation with RSA | | Key decapsulation with RSA (KTS-Decapsulation) |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | decapsulation, generated by the module | ~152, ~200 | | | | |
| RSA Key Decapsulation Private Key (Established) | Key (including intermediate values) for performing RSA key decapsulation, established by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key decapsulation with RSA (KTS-Decapsulation) |
| RSA Key Encapsulation Public Key (Generated) | Key for performing RSA key decapsulation, generated by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Public - PSP | Key pair generation with RSA | | Key encapsulation with RSA (KTS-Encapsulation) |
| RSA Key Encapsulation Public Key (Established) | Keys for performing RSA key encapsulation, established by the module | 2048, 3072, 4096 or 8192 bits - 112, 128, ~152, ~200 | RSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key encapsulation with RSA (KTS-Encapsulation) |
| ECDSA Digital Signature Private Keys (Generated) | Keys (including intermediate values) for performing ECDSA digital signature generation, generated by the module | P-224, P-256, P-384, P-521 - 112, 128, 192, 256 | ECDSA Private - CSP | Key pair generation with RSA | | Digital Signature generation with ECDSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| ECDSA Digital Signature Private Keys (Established) | Keys for performing ECDSA digital signature generation, established by the module | P-224, P-256, P-384, P-521 - 112, 128, 192, 256 | ECDSA Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature generation with ECDSA |
| ECDSA Digital Signature Public Keys (Generated) | Keys for performing ECDSA digital signature validation, generated by the module | P-224, P-256, P-384, P-521 - 112, 128, 192, 256 | ECDSA Public - PSP | Key pair generation with ECDSA | | Digital Signature verification with ECDSA |
| ECDSA Digital Signature Public Keys (Established) | Keys for performing ECDSA digital signature validation, established by the module | P-224, P-256, P-384, P-521 - 112, 128, 192, 256 | ECDSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with ECDSA |
| EDDSA Digital Signature Private Keys (Generated) | Keys (including intermediate values) for performing EDDSA digital signature generation, generated by the module | ED-255, ED-448 - 128, 224 | EDDSA Private - CSP | Key pair generation with EDDSA | | Digital Signature generation with EDDSA |
| EDDSA Digital Signature Private Keys (Established) | Keys (including intermediate values) for | ED-255, ED-448 - 128, 224 | EDDSA Private - CSP | | Key unwrapping with AES-KW or | Digital Signature generation |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | performing EDDSA digital signature generation, established by the module | | | | AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | with EDDSA |
| EDDSA Digital Signature Public Keys (generated) | Keys for performing EDDSA digital signature validation, generated by the module | ED-255, ED-448 - 128, 224 | EDDSA Public - PSP | Key pair generation with EDDSA | | Digital Signature verification with EDDSA |
| EDDSA Digital Signature Public Keys (Established) | Keys for performing EDDSA digital signature validation, established by the module | ED-255, ED-448 - 128, 224 | EDDSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with EDDSA |
| LMS Digital Signature Private Keys (Generated) | Keys (including intermediate values) for performing LMS digital signature generation, generated by the module | minimum of 416-bits for LMS/LMOTS where M = 24, N = 24 and minimum of 480-bits for LMS/LMOTS where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits | LMS Private - CSP | Key pair generation with LMS | | Digital Signature generation with LMS |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | for M/N = 32 | | | | |
| LMS Digital Signature Public Keys (Generated) | Public Keys (including intermediate values) for performing LMS digital signature verification, generated by the module | 384-bits for LMS/LMOTS where M = 24, N = 24 and 448-bits for LMS/LMOTS where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits for M/N = 32 | LMS Public - PSP | Key pair generation with LMS | | Digital Signature verification with LMS |
| LMS Digital Signature Public Keys (Established) | Keys for performing LMS digital signature validation, established by the module | 384-bits for LMS/LMOTS where M = 24, N = 24 and 448-bits for LMS/LMOTS where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits for M/N = 32 | LMS Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with LMS |
| ML-DSA Seed | Seed used in ML-DSA Key Generation | 256 bits - 256 | Seed - CSP - CSP | Random number generation with Hash DRBG | | Key pair generation with ML-DSA |
| ML-DSA Digital Signature | Keys (including intermediate values) for | 2560, 4032 or 4896 bytes - | ML-DSA Private - CSP | Key pair generation with ML-DSA | | Digital Signature generation with ML- |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| Private Keys (Generated) | performing ML-DSA digital signature generation, generated by the module | 128, 192, or 256-bits | | | | DSA (External Interface, Pure, Hedged) Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM) Digital Signature generation with ML-DSA (Internal Interface, Hedged, External Mu) |
| ML-DSA Digital Signature Private Keys (Established) | Keys (including intermediate values) for performing ML-DSA digital signature generation, established by the module | 2560, 4032 or 4896 bytes - 128, 192, or 256-bits | ML-DSA Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature generation with ML-DSA (External Interface, Pure, Hedged) Digital Signature generation with ML- |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | | | | | DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with ML-DSA (External Interface, PreHash, Hedged, PHM) Digital Signature generation with ML-DSA (Internal Interface, Hedged, External Mu) |
| ML-DSA Digital Signature Public Keys (Generated) | Keys for performing ML-DSA digital signature validation, generated by the module | 1312, 1952 or 2592 bytes - 128, 192, or 256-bits | ML-DSA Public - PSP | Key pair generation with ML-DSA | | Digital Signature verification with ML-DSA (External Interface, Pure) Digital Signature verification with ML-DSA (External Interface, PreHash, Message) Digital Signature verification with ML-DSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | | | | | (External Interface, PreHash, PHM) |
| ML-DSA Digital Signature Public Keys (Established) | Keys for performing ML-DSA digital signature validation, established by the module | 1312, 1952 or 2592 bytes - 128, 192, or 256-bits | ML-DSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with ML-DSA (External Interface, Pure) Digital Signature verification with ML-DSA (External Interface, PreHash, Message) Digital Signature verification with ML-DSA (External Interface, PreHash, PHM) |
| ML-KEM Seed | Seed used in ML-KEM Key Generation | 256 bits - 256 | Seed - CSP - CSP | Random number generation with Hash DRBG | | Key pair generation with ML-KEM |
| ML-KEM Shared Secret | Shared secret K obtained from ciphertext | 256 bits - 256 | Shared secret - CSP - CSP | Key encapsulation with ML-KEM | | |
| ML-KEM Key Decapsulation Private Key (Generated) | Keys (including intermediate values) for performing ML-KEM key | 1632, 2400 or 3168 bytes - 128, 192, or 256-bits | ML-KEM Private - CSP | Key pair generation with ML-KEM | | Key decapsulation with ML-KEM |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | decapsulation, generated by the module | | | | | |
| ML-KEM Key Decapsulation Private Key (Established) | Keys (including intermediate values) for performing ML-KEM key decapsulation, established by the module | 1632, 2400 or 3168 bytes - 128, 192, or 256-bits | ML-KEM Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key decapsulation with ML-KEM |
| ML-KEM Key Encapsulation Public Key (Generated) | Keys for performing ML-KEM key encapsulation, generated by the module | 800, 1184 or 1568 bytes - 128, 192, or 256-bits | ML-KEM Public - PSP | Key pair generation with ML-KEM | | Key encapsulation with ML-KEM |
| ML-KEM Key Encapsulation Public Key (Established) | Keys for performing ML-KEM key encapsulation, established by the module | 800, 1184 or 1568 bytes - 128, 192, or 256-bits | ML-KEM Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key encapsulation with ML-KEM |
| SLH-DSA Seed | Seed for SLH-DSA Key Pair Generation | 384, 576, or 768 - 128, 192, or 256 | Seed - CSP - CSP | Random number generation with Hash DRBG | | Key pair generation with SLH-DSA |
| SLH-DSA Digital Signature Private Keys (Generated) | Keys (including intermediate values) for performing | 512, 768, 1024 bits - 128, 192, or 256 | SLH-DSA Private - CSP | Key pair generation with SLH-DSA | | Digital Signature generation with SLH-DSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | SLH-DSA digital signature generation, generated by the module | | | | | (External Interface, Pure, Hedged) Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, PHM) |
| SLH-DSA Digital Signature Private Keys (Established) | Keys (including intermediate values) for performing SLH-DSA digital signature generation, established by the module | 512, 768, 1024 bits - 128, 192, or 256-bits | SLH-DSA Private - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature generation with SLH-DSA (External Interface, Pure, Hedged) Digital Signature generation with SLH-DSA (External Interface, PreHash, Hedged, Message) Digital Signature generation with SLH-DSA |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | | | | | (External Interface, PreHash, Hedged, PHM) |
| SLH-DSA Digital Signature Public Keys (Generated) | Keys for performing SLH-DSA digital signature validation, generated by the module | 256, 384, 512 bits - 128, 192, or 256-bits | SLH-DSA Public - PSP | Key pair generation with SLH-DSA | | Digital Signature verification with SLH-DSA (External Interface, Pure) Digital Signature verification with SLH-DSA (External Interface, PreHash, Message) Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) |
| SLH-DSA Digital Signature Public Keys (Established) | Keys for performing SLH-DSA digital signature validation, established by the module | 256, 384, 512 bits - 128, 192, or 256-bits | SLH-DSA Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with SLH-DSA (External Interface, Pure) Digital Signature verification with SLH-DSA (External Interface, PreHash, Message) |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | | | | | Digital Signature verification with SLH-DSA (External Interface, PreHash, PHM) |
| KBKDF Key Derivation Keys (Generated) | Keys for performing key-based key derivation, generated by the module | 128-4096 bits - 128, 192, or 256-bits | KBKDF - CSP | Random number generation with Hash DRBG | | Key derivation with KBKDF |
| KBKDF Key Derivation Keys (Established) | Keys for performing key-based key derivation, established by the module | 128-4096 bits - 128, 192, or 256-bits | KBKDF - CSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Key derivation with KBKDF |
| KBKDF Derived Key | Keys produced by the KBKDF mechanism | Contingent on key derivation key and output length - 128, 192, or 256-bits | KBKDF - CSP | Key derivation with KBKDF | | |
| HSS Digital Signature Private Keys (Generated) | HSS private signing key represented as a single root secret (SEED  I index) per RFC 8554 Appendix A | minimum of 416-bits for LMS/LMOTS where M = 24, N = 24 and minimum of 480-bits for | HSS Private - CSP | Key pair generation with HSS | | Digital Signature generation with HSS |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | LMS/LMOTS where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits for M/N = 32 | | | | |
| HSS Digital Signature Public Keys (Generated) | Keys for performing HSS digital signature validation, generated by the module | 384-bits where M = 24, N = 24 and 448-bits where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits for M/N = 32 | HSS Public - PSP | Key pair generation with HSS | | Digital Signature verification with HSS |
| HSS Digital Signature Public Keys (Established) | Keys for performing HSS digital signature validation, established by the module | 384-bits where M = 24, N = 24 and 448-bits where M = 32, N = 32 - 192-bits for M/N = 24 and 256-bits for M/N = 32 | HSS Public - PSP | | Key unwrapping with AES-KW or AES-KWP Key decapsulation with RSA (KTS-Decapsulation) | Digital Signature verification with HSS |
| ECDH Recipient Public Key | Keys passed from the provider to the recipient to derive a shared secret | P-224, P-256, P-384, P-521 - 112, 128, 192, 256 | ECDH Public - CSP | | | Elliptic curve Diffie-Hellman key agreement |
| ECDH Provider Private Key | Keys possessed by the | P-224, P-256, P-384, P- | ECDH Private - CSP | Elliptic curve Diffie- | | Elliptic curve Diffie- |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | provider that are used to derive a shared secret | 521 - 112, 128, 192, 256 | | Hellman key agreement | | Hellman key agreement |
| ECC Shared Secret | Secret material produced during the ECDH key agreement process | 256 bits - 256 | ECDH - CSP | | Elliptic curve Diffie-Hellman key agreement | |
| ECDH Derived Key | A key produced at the end of the ECDH key agreement process | 256 bits - 256 | ECDH - CSP | | Elliptic curve Diffie-Hellman key agreement | |
| KDF ANS 9.63 Derived Key | Keying material derived from a shared secret using the ANS 9.63 key derivation scheme | 128-4096 bits - 192, 256 | ANS 9.63 KDF - CSP | | Elliptic curve Diffie-Hellman key agreement | |
| SP800-56Cr2 Derived Key | Keying material derived from a shared secret using the SP800-56Cr2 key derivation scheme | 256 bits - 256 | SP800-56Cr2 - CSP | | Elliptic curve Diffie-Hellman key agreement | |

Table 20: SSP Table 1

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| TKA Wrapping Key for ATP0 (TKAWK0) | | ATP0 FRAM:Plaintext<br>ATP1 FRAM:Encrypted | | Tamper Zeroization | Tamper Key for ATP0 (TKA0):Decrypts |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| Tamper Key for ATP0 (TKA0) | | ATP0 SRAM:Plaintext<br>ATP0 FRAM:Encrypted | From first initialization until a tamper event is detected by ATP0 | Tamper Zeroization | Tamper Key for HSM (TKH):Decrypts<br>Tamper Key for HaaP (TKU):Decrypts<br>Zeroization Key for HaaP (ZKU):Decrypts |
| TKA Wrapping Key for ATP1 (TKAWK1) | | ATP1 FRAM:Plaintext<br>ATP0 FRAM:Encrypted | | Tamper Zeroization | Tamper Key for ATP1 (TKA1):Decrypts |
| Tamper Key for ATP1 (TKA1) | | ATP1 SRAM:Plaintext<br>ATP1 FRAM:Encrypted | From first initialization until a tamper event is detected by ATP1 | Tamper Zeroization | Tamper Key for HSM (TKH):Decrypts<br>Tamper Key for HaaP (TKU):Decrypts<br>Zeroization Key for HaaP (ZKU):Decrypts |
| Tamper Key for HSM (TKH) | | ATP0 FRAM:Encrypted<br>ATP1 FRAM:Encrypted<br>ARTHUR FRAM:Encrypted | From first initialization until a tamper event is detected | Tamper Zeroization | |
| Tamper Key for HaaP (TKU) | | ATP0 FRAM:Encrypted<br>ATP1 FRAM:Encrypted<br>ARTHUR FRAM:Encrypted | From first initialization until a tamper event is detected | Tamper Zeroization | Zeroization Key for HaaP (ZKU):Decrypts |
| Zeroization Key for HaaP (ZKU) | | ATP0 FRAM:Encrypted<br>ATP1 FRAM:Encrypted | From first initialization until zeroization is performed | Zeroize module service<br>Manual Zeroization | Tenant Master Key (TMK):Decrypts |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | ARTHUR FRAM:Encrypted | or a tamper event is detected | | |
| Tenant Master Key (TMK) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | | Zeroization Key for HaaP (ZKU):Decrypts |
| DRBG Entropy Input | | ARTHUR DRAM:Encrypted | From the DRBG function call until the DRBG seed is generated | Tamper Zeroization Zeroize module service Manual Zeroization Automatic | |
| DRBG V | | ARTHUR DRAM:Encrypted | From the instantiation of the DRBG until reinstantiation or reseeding occurs | Tamper Zeroization Zeroize module service Manual Zeroization Automatic | DRBG Seed:Derived From |
| DRBG C | | ARTHUR DRAM:Encrypted | From the instantiation of the DRBG until reinstantiation or reseeding occurs | Tamper Zeroization Zeroize module service Manual Zeroization Automatic | DRBG V:Derived From |
| DRBG Seed | | ARTHUR DRAM:Encrypted | From the instantiation of the DRBG until reinstantiati | Tamper Zeroization Zeroize module service | DRBG Entropy Input:Derived From |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | on or reseeding | Manual Zeroization Automatic | |
| C4A_TA_ECDSA | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| C4A_TA_HSS | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| C4A_RCA | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| C4A_SCA | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| QASM_IA_HSS (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_IA_HSS (Public Key):Paired With |
| QASM_IA_HSS (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_IA_HSS (Private Key):Paired With |
| QASM_IA_ECC (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_IA_ECC (Public Key):Paired With |
| QASM_IA_ECC (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is | Tamper Zeroization | QASM_IA_ECC (Private Key):Paired With |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | performed or a tamper event occurs | | |
| QASM_KE_ECC (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_KE_ECC (Public Key):Paired With |
| QASM_KE_ECC (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_KE_ECC (Private Key):Paired With |
| QASM_AA_HSS (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_AA_HSS (Public Key):Paired With |
| QASM_AA_HSS (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_AA_HSS (Private Key):Paired With |
| QASM_AA_ECC (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper | Tamper Zeroization | QASM_AA_ECC (Public Key):Paired With |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | event occurs | | |
| QASM_AA_ECC (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_AA_ECC (Private Key):Paired With |
| C4A_RCA_COM | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| C4A_SCA_COM | | ARTHUR FRAM:Encrypted | | Tamper Zeroization | |
| QASM_SCA_COM (Private Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_SCA_COM (Public Key):Paired With |
| QASM_SCA_COM (Public Key) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | QASM_SCA_COM (Private Key):Paired With |
| Owner Public Key(C4A) | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Tamper Zeroization | |
| New Owner Public Key | | ARTHUR FRAM:Encrypted | From ownership transfer until | Tamper Zeroization | Owner Public Key(C4A):Replaces |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | zeroization is performed or a tamper event occurs | | |
| Security Officer Public Key | | ARTHUR FRAM:Encrypted | From first initialization until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| AES Symmetric Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| AES Symmetric Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES MoFN (Import) MoFN (Export) | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| AES-KW Symmetric Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| AES-KW Symmetric Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES MoFN (Import) MoFN (Export) | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| AES-GCM Symmetric Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| AES-GCM Symmetric Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper | Zeroize module service Manual Zeroization Tamper | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | AES Wrap with RSA Unpack with AES Pack with AES MoFN (Import) MoFN (Export) | | event occurs | Zeroization | |
| HMAC Symmetric Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES MoFN (Import) MoFN (Export) | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| RSA Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| RSA Digital Signature | Unwrap with AES | ARTHUR DRAM:Encrypted | From establishment until | Zeroize module service | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| Private Keys (Established) | Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | | zeroization is performed or a tamper event occurs | Manual Zeroization Tamper Zeroization | |
| RSA Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| RSA Digital Signature Public Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| RSA Key Decapsulation Private Key (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed | Zeroize module service Manual Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | or a tamper event occurs | Tamper Zeroization | |
| RSA Key Decapsulation Private Key (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| RSA Key Encapsulation Public Key (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| RSA Key Encapsulation Public Key (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| ECDSA Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| ECDSA Digital Signature Private Keys (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| ECDSA Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| ECDSA Digital Signature Public Keys (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | Unpack with AES Pack with AES | | | | |
| EDDSA Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| EDDSA Digital Signature Private Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| EDDSA Digital Signature Public Keys (generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| EDDSA Digital Signature Public Keys (Established) | Unwrap with AES Unwrap with RSA | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed | Zeroize module service Manual Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | | or a tamper event occurs | Tamper Zeroization | |
| LMS Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| LMS Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| LMS Digital Signature Public Keys (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| ML-DSA Seed | | ARTHUR DRAM:Plaintext | Until cipher handle is released or module is reset | Tamper Zeroization<br>Zeroize module service<br>Manual Zeroization<br>Automatic | |
| ML-DSA Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| ML-DSA Digital Signature Private Keys (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |
| ML-DSA Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| ML-DSA Digital Signature Public Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| ML-KEM Seed | | ARTHUR DRAM:Plaintext | Until cipher handle is released or module is reset | Tamper Zeroization Zeroize module service Manual Zeroization Automatic | |
| ML-KEM Shared Secret | | ARTHUR DRAM:Plaintext | Until cipher handle is released or module is reset | Zeroize module service Manual Zeroization Tamper Zeroization | ML-KEM Key Decapsulation Private Key (Generated):Derived From ML-KEM Key Decapsulation Private Key (Established):Derived From ML-KEM Key Encapsulation Public Key (Generated):Derived From ML-KEM Key Encapsulation Public Key (Established):Derived From |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| ML-KEM Key Decapsulation Private Key (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | ML-KEM Key Encapsulation Public Key (Generated):Paired With |
| ML-KEM Key Decapsulation Private Key (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | ML-KEM Key Encapsulation Public Key (Established):Paired With |
| ML-KEM Key Encapsulation Public Key (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | ML-KEM Key Decapsulation Private Key (Generated):Paired With |
| ML-KEM Key Encapsulation Public Key (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | ML-KEM Key Encapsulation Private Key (Established):Paired With |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | Unpack with AES<br>Pack with AES | | | | |
| SLH-DSA Seed | | ARTHUR DRAM:Encrypted | Until cipher handle is released or module is reset | Tamper Zeroization<br>Zeroize module service<br>Manual Zeroization<br>Automatic | |
| SLH-DSA Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | SLH-DSA Digital Signature Public Keys (Generated):Paired With |
| SLH-DSA Digital Signature Private Keys (Established) | Unwrap with AES<br>Unwrap with RSA<br>Wrap with AES<br>Wrap with RSA<br>Unpack with AES<br>Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service<br>Manual Zeroization<br>Tamper Zeroization | SLH-DSA Digital Signature Public Keys (Established):Paired With |
| SLH-DSA Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is | Zeroize module service<br>Manual Zeroizatio | SLH-DSA Digital Signature Private Keys (Generated):Paired With |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | performed or a tamper event occurs | n Tamper Zeroization | |
| SLH-DSA Digital Signature Public Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | SLH-DSA Digital Signature Private Keys (Established):Paired With |
| KBKDF Key Derivation Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| KBKDF Key Derivation Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | with AES | | | | |
| KBKDF Derived Key | Wrap with AES Wrap with RSA Pack with AES | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| HSS Digital Signature Private Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | HSS Digital Signature Public Keys (Generated):Paired With |
| HSS Digital Signature Public Keys (Generated) | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | HSS Digital Signature Private Keys (Generated):Paired With |
| HSS Digital Signature Public Keys (Established) | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | HSS Digital Signature Private Keys (Established):Paired With |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| ECDH Recipient Public Key | Import Recipient Public Key | ARTHUR DRAM:Plaintext | From input until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| ECDH Provider Private Key | | ARTHUR DRAM:Encrypted | From generation until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| ECC Shared Secret | | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Automatic | |
| ECDH Derived Key | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |
| KDF ANS 9.63 Derived Key | Unwrap with AES | ARTHUR DRAM:Encrypted | From establishment until | Zeroize module service | |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | | zeroization is performed or a tamper event occurs | Manual Zeroization Tamper Zeroization | |
| SP800-56Cr2 Derived Key | Unwrap with AES Unwrap with RSA Wrap with AES Wrap with RSA Unpack with AES Pack with AES | ARTHUR DRAM:Encrypted | From establishment until zeroization is performed or a tamper event occurs | Zeroize module service Manual Zeroization Tamper Zeroization | |

Table 21: SSP Table 2

There is currently no official source for the strength and size values of MCE (McEliece).

## 9.5 Transitions

SHA-1 is disallowed for digital signature generation. When used for digital signature verification, SHA-1 is allowed for legacy use. The use of SHA-1 is deprecated through December 31, 2030, for applying protection in non-digital signature applications and disallowed thereafter. The use of SHA-1 is acceptable for processing already-protected information through December 31, 2030, and allowed for legacy use thereafter.

The use of a key with 112-bit security strength is acceptable for applying protection and for processing already-protected information through December 31, 2030. After that date, the use of a key with 112-bit security strength will be disallowed for applying new protection and will be permitted only for legacy use in processing already-protected information.

# 10 Self-Tests

While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module does not return control to the calling application until the tests are completed.

The error logging functionality provided by the module includes the most recent error event.

## 10.1 Pre-Operational Self-Tests

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details |
|---|---|---|---|---|---|
| AES-GCM (A5631) | 256-bit key | Authentication | SW/FW Integrity | Implicit; Successful boot of module implies integrity has been verified. | FSBL then verifies integrity of the operational image by verifying its GMAC tag during AES-GCM decryption of the firmware. If the decryption succeeds, thus confirming the GMAC tag was verified, the module firmware is loaded and begins executing. |
| EDC | CRC-32 | Error Detection Code | SW/FW Integrity | Implicit; Successful boot of module implies integrity has been verified. | When the module powers up, the FSBL performs a CRC-32 on itself. |

Table 22: Pre-Operational Self-Tests

## 10.2 Conditional Self-Tests

The module performs all required cryptographic algorithm self-tests at power-on. The module implements a service for an operator to additionally invoke the self-tests on demand. This is noted as “Self-test” in the Approved Services table.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-GCM (A5631) | 128, 192, 256 bit key; Decryption | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Symmetric operation | Test runs at power on prior to its first use. |
| AES-GCM (A5631) | 128, 192, 256 bit | KAT | CAST | Implicit; Successful | Symmetric operation | Test runs at power |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | key; Encryption | | | boot of module implies KAT was passed. | | on prior to its first use. |
| AES-GCM (A6119) | 256 bit key; Decryption | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Symmetric operation | Test runs at power on prior to its first use. |
| AES-GCM (A6119) | 256 bit key; Encryption | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Symmetric operation | Test runs at power on prior to its first use. |
| AES-ECB (A5631) | 128, 192, 256 bit keys; Decryption | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Symmetric operation | Test runs at power on prior to its first use. |
| AES-ECB (A5631) | 128, 192, 256 bit keys; Encryption | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Symmetric operation | Test runs at power on prior to its first use. |
| SHA-1 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA2-224 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA2-256 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| SHA2-384 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA2-512 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA2-512/224 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA2-512/256 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA3-224 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA3-256 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA3-384 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message digest | Test runs at power on prior to its first use. |
| SHA3-512 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module | Message digest | Test runs at power on prior to its first use. |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | implies KAT was passed. | | |
| HMAC-SHA-1 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA2-224 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA2-256 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA2-384 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA2-512 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA3-224 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA3-256 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Message authentication | Test runs at power on prior to its first use. |
| HMAC-SHA3-512 (A5631) | N/A | KAT | CAST | Implicit; Successful boot of | Message authentication | Test runs at power on prior to |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | module implies KAT was passed. | | its first use. |
| DRBG Instantiate | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Compliant with SP800 90Arev1 | Test runs at power on prior to its first use. |
| DRBG Generate | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Compliant with SP800 90Arev1 | Test runs at power on prior to its first use. |
| DRBG Reseed | N/A | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Compliant with SP800 90Arev1 | Test runs at power on prior to its first use. |
| ECDSA SigGen (FIPS186-5) (A5631) | P-224, P-256, P-384 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature generation | Test runs at power on prior to its first use. |
| ECDSA SigVer (FIPS186-5) (A5631) | P-224, P-256, P-384 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature verification | Test runs at power on prior to its first use. |
| ECDSA KeyGen (FIPS186-5) (A5631) | P-224, P-256, P-384, P-521 | PCT | PCT | HSM_RET_OK is returned | Signature generation & verification | ECDSA key pair generation |
| EDDSA SigGen (FIPS186-5) (A5631) | Ed448, Ed25519 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature generation | Test runs at power on prior to its first use. |
| EDDSA SigVer (FIPS186-5) (A5631) | Ed448, Ed25519 | KAT | CAST | Implicit; Successful boot of module | Signature verification | Test runs at power on prior to |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | implies KAT was passed. | | its first use. |
| EDDSA KeyGen (FIPS186-5) (A5631) | Ed448, Ed25519 | PCT | PCT | HSM_RET_OK is returned | Signature generation & verification | EdDSA key pair generation |
| RSA SigGen (FIPS186-5) (A5631) | Modulus Size: 2048, 3072, 4096; PSS, PKCS1-v1.5 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature generation | Test runs at power on prior to its first use. |
| RSA SigVer (FIPS186-5) (A5631) | Modulus Size: 2048, 3072, 4096; PSS, PKCS1-v1.5 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature verification | Test runs at power on prior to its first use. |
| RSA KeyGen (FIPS186-5) (A5631) | Modulus Size: 2048, 3072, 4096, 8192 | PCT | PCT | HSM_RET_OK is returned | Signature generation & verification, encapsulation & decapsulation | RSA key pair generation |
| RSA SigGen (FIPS186-5) (A6409) | Modulus Size: 2048, 3072, 4096; PSS, PKCS1-v1.5 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature generation (CSU assisted RSA) | Test runs at power on prior to its first use. |
| RSA SigVer (FIPS186-5) (A6409) | Modulus Size: 2048, 3072, 4096; PSS, PKCS1-v1.5 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature verification (CSU assisted RSA) | Test runs at power on prior to its first use. |
| RSA SigGen | Modulus Size: 2048, | KAT | CAST | Implicit; Successful boot of | Signature generation | Test runs at power on prior to |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| (FIPS186-5) (A6410) | 3072, 4096; PSS, PKCS1-v1.5 | | | module implies KAT was passed. | (FPGA assisted RSA) | its first use. |
| RSA SigVer (FIPS186-5) (A6410) | Modulus Size: 2048, 3072, 4096; PSS, PKCS1-v1.5 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Signature verification (FPGA assisted RSA) | Test runs at power on prior to its first use. |
| LMS SigGen (A5631) | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation | Test runs at power on prior to its first use. |
| LMS SigVer (A5631) | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature verification | Test runs at power on prior to its first use. |
| LMS KeyGen (A5631) KAT | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation & verification | Test runs at power on prior to its first use. |
| LMS KeyGen (A5631) PCT | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | PCT | PCT | HSM_RET_OK is returned | Signature generation & verification | LMS key pair generation |
| HSS SigGen | Hash: SHA256, | KAT | CAST | Implicit; Successful | Signature generation | Test runs at power |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | | | boot of module implies KAT was passed | | on prior to its first use. |
| HSS SigVer | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature verification | Test runs at power on prior to its first use. |
| HSS KeyGen KAT | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation & verification | Test runs at power on prior to its first use. |
| HSS KeyGen PCT | Hash: SHA256, SHAKE; Size: 24, 32 bytes; Height: 5, 10, 15, 20, 25 | PCT | PCT | HSM_RET_OK is returned | Signature generation & verification | HSS key pair generation |
| ML-DSA SigGen (A5631) | ML-DSA-44, ML-DSA-65, ML-DSA-87 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation; including two rejection sampling loop paths (checks on z and r0); performed using the external implementations. Both "Pure" and "PreHash" paths are tested. | Test runs at power on prior to its first use. |
| ML-DSA SigVer (A5631) | ML-DSA-44, ML-DSA-65, | KAT | CAST | Implicit; Successful boot of | Signature verification; performed using | Test runs at power on prior to |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | ML-DSA-87 | | | module implies KAT was passed | the external implementations. Both "Pure" and "PreHash" paths are tested. | its first use. |
| ML-DSA KeyGen (A5631) KAT | ML-DSA-44, ML-DSA-65, ML-DSA-87 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | ML-DSA Key Generation | Test runs at power on prior to its first use. |
| ML-DSA KeyGen (A5631) PCT | ML-DSA-44, ML-DSA-65, ML-DSA-87 | PCT | PCT | HSM_RET_OK returned | Signature generation & verification | ML-DSA key pair generation |
| SLH-DSA SigGen (A5631) | SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation; performed using the external implementations. Both "Pure" and "PreHash" paths are tested. | Test runs at power on prior to its first use. |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | | | | | |
| SLH-DSA SigVer (A5631) | SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature verification; performed using the external implementations. Both "Pure" and "PreHash" paths are tested. | Test runs at power on prior to its first use. |
| SLH-DSA KeyGen (A5631) KAT | SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | Signature generation & verification | SLH-DSA key pair generation |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | | | | | |
| SLH-DSA KeyGen (A5631) PCT | SLH-DSA-SHA2-128f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-256f, SLH-DSA-SHA2-256s, | PCT | PCT | HSM_RET_OK returned | Signature generation & verification | SLH-DSA key pair generation |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256f, SLH-DSA-SHAKE-256s | | | | | |
| KAS-ECC Sp800-56Ar3 (A5631) | P-224 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | EC Diffie-Hellman key agreement | Test runs at power on prior to its first use. |
| KDA OneStep SP800-56Cr2 (A5631) | Derived Key Length: 256 bits | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key Derivation | Test runs at power on prior to its first use. |
| KDF ANS 9.63 (A5631) | Field Size: 224 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key Derivation | Test runs at power on prior to its first use. |
| KDF SP800-108 (A5631) | HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key Derivation | Test runs at power on prior to its first use. |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 | | | | | |
| AES-KW (A5631) | 128, 192, 256 bit KEK; Key wrap | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key wrap | Test runs at power on prior to its first use. |
| AES-KW (A5631) | 128, 192, 256 bit KEK; Key unwrap | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key unwrap | Test runs at power on prior to its first use. |
| AES-KWP (A5631) | 128, 192, 256 bit KEK; Key wrap | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key wrap | Test runs at power on prior to its first use. |
| AES-KWP (A5631) | 128, 192, 256 bit KEK; Key unwrap | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | Key unwrap | Test runs at power on prior to its first use. |
| KTS-IFC (A5631) | Modulus Size: 2048 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed. | key encapsulate and unencapsulate tested separately | Test runs at power on prior to its first use. |
| KTS-IFC (A6409) | Modulus Size: 2048 | KAT | CAST | Implicit; Successful boot of | key encapsulate and unencapsulate | Test runs at power on prior to |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | module implies KAT was passed. | tested separately | its first use. |
| KTS-IFC (A6410) | Modulus Size: 2048 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | key encapsulate and unencapsulate tested separately | Test runs at power on prior to its first use. |
| ML-KEM EncapDecap (A5631) | ML-KEM-512, ML-KEM-768, ML-KEM-1024 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | key encapsulate and unencapsulate tested separately | Test runs at power on prior to its first use. |
| ML-KEM KeyGen (A5631) KAT | ML-KEM-512, ML-KEM-768, ML-KEM-1024 | KAT | CAST | Implicit; Successful boot of module implies KAT was passed | ML-KEM Key Generation | Test runs at power on prior to its first use. |
| ML-KEM KeyGen (A5631) PCT | ML-KEM-512, ML-KEM-768, ML-KEM-1024 | PCT | PCT | HSM_RET_OK is returned | encapsulation and decapsulation | ML-KEM key pair generation |
| ESV-RCT | Cutoff value = 6 | Fault-detection test | CAST | Successful seeding of SP 800-90A DRBG | SP 800-90B 4.4.1 Repetition Count Test | Upon seeding or reseeding SP 800-90A DRBG |
| ESV-APT | Cutoff value = 31 | Fault-detection test | CAST | Successful seeding of SP 800-90A DRBG | SP 800-90B 4.4.2 Adaptive Proportion Test | Upon seeding or reseeding SP 800-90A DRBG |

Table 23: Conditional Self-Tests

## 10.3 Periodic Self-Test Information

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-GCM (A5631) | Authentication | SW/FW Integrity | Power-on, Reboot | Manual |
| EDC | Error Detection Code | SW/FW Integrity | Power-on, Reboot | Manual |

Table 24: Pre-Operational Periodic Information

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-GCM (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-GCM (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-GCM (A6119) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-GCM (A6119) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-ECB (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-ECB (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA-1 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA2-224 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA2-256 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA2-384 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA2-512 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| SHA2-512/224 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA2-512/256 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA3-224 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA3-256 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA3-384 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SHA3-512 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA-1 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA2-224 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA2-256 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA2-384 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA2-512 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA3-224 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA3-256 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HMAC-SHA3-512 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| DRBG Instantiate | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| DRBG Generate | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| DRBG Reseed | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ECDSA SigGen (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ECDSA SigVer (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ECDSA KeyGen (FIPS186-5) (A5631) | PCT | PCT | On demand | Manually |
| EDDSA SigGen (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| EDDSA SigVer (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| EDDSA KeyGen (FIPS186-5) (A5631) | PCT | PCT | On demand | Manually |
| RSA SigGen (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| RSA SigVer (FIPS186-5) (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| RSA KeyGen (FIPS186-5) (A5631) | PCT | PCT | On demand | Manually |
| RSA SigGen (FIPS186-5) (A6409) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| RSA SigVer (FIPS186-5) (A6409) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| RSA SigGen (FIPS186-5) (A6410) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| RSA SigVer (FIPS186-5) (A6410) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| LMS SigGen (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| LMS SigVer (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| LMS KeyGen (A5631) KAT | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| LMS KeyGen (A5631) PCT | PCT | PCT | On demand | Manually |
| HSS SigGen | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HSS SigVer | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| HSS KeyGen KAT | KAT | CAST | Power-on, reboot, or after | Automatic; Every 12 hours |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| | | | periodic interval is reached | |
| HSS KeyGen PCT | PCT | PCT | On demand | Manually |
| ML-DSA SigGen (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-DSA SigVer (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-DSA KeyGen (A5631) KAT | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-DSA KeyGen (A5631) PCT | PCT | PCT | On demand | Manually |
| SLH-DSA SigGen (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SLH-DSA SigVer (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| SLH-DSA KeyGen (A5631) KAT | KAT | CAST | On demand | Manually |
| SLH-DSA KeyGen (A5631) PCT | PCT | PCT | On demand | Manually |
| KAS-ECC Sp800-56Ar3 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KDA OneStep SP800-56Cr2 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KDF ANS 9.63 (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KDF SP800-108 (A5631) | KAT | CAST | Power-on, reboot, or after | Automatic; Every 12 hours |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| | | | periodic interval is reached | |
| AES-KW (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-KW (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-KWP (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| AES-KWP (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KTS-IFC (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KTS-IFC (A6409) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| KTS-IFC (A6410) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-KEM EncapDecap (A5631) | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-KEM KeyGen (A5631) KAT | KAT | CAST | Power-on, reboot, or after periodic interval is reached | Automatic; Every 12 hours |
| ML-KEM KeyGen (A5631) PCT | PCT | PCT | On demand | Manually |
| ESV-RCT | Fault-detection test | CAST | On demand | Manually |
| ESV-APT | Fault-detection test | CAST | On demand | Manually |

Table 25: Conditional Periodic Information

The module's periodic self-tests are performed repeatedly on an interval of 12 hours.

## 10.4 Error States

| Name | Description | Conditions | Recovery Method | Indicator |
|---|---|---|---|---|
| Soft Error | The module enter this state in the event of a failure of any self-test or the pre-operational integrity test. While in the Soft Error state, all cryptographic functions as well as the module's data output interface is inhibited. The module can be recovered by rebooting and successfully passing all self-tests | The failure of any CAST or pre-operational integrity test | Reboot | GPIO status output pins; error message |
| Hard Error | The module enters the Hard Error state when a tamper event is detected. While in the Hard Error state, all cryptographic functions as well as all input and output is inhibited. | Tamper event detection | N/A | The module is non-operational. |

Table 26: Error States

# 11 Life-Cycle Assurance

Git is the configuration management system for the module. All versions of all configuration items are uniquely identified using Git commits. Changes to the code must first be submitted to Gerrit for code review before such changes are made to the main code branches.

## 11.1 Installation, Initialization, and Startup Procedures

During its manufacturing process, the QASM is assembled in a QxEDGE or QxHSM. Before the module can be used in the FIPS validated state, the following condensed steps are to be followed. Note, the steps shown are based on the QASM being integrated in a QxEDGE but is applicable to the QxHSM as well. A more detailed procedure for proper installation, initialization, and startup of the module is documented within the “QxEdge getting started”, “Platform Installation Guide”, and “QxEDGE Modes” documents:

1. Connect the power cord(s) and the ethernet cables.
2. Power up the QxEDGE by pressing the front panel push button. This will power-up the QASM.
3. Configure the QxEDGE using one of two methods:
    a. Use Crytpo4A’s Universal Configuration Application (UCA) Universal Configuration Application
    b. Load configuration files from a server and installed them on the QxEDGE which will configure itself automatically.
4. Optionally, the end user can choose to transfer ownership of the device from Crypto4A (the default owner of all manufactured devices) to themselves using the following out-of-band ownership transfer procedure:
    a. The end user generates a private/public keypair.
    b. The end user generates an ownership transfer request that identifies the newly generated public key as the new Owner Public Key and submits that to Crypto4A.
    c. Crypto4A signs the ownership transfer request using their existing Owner Private Key and returns that signature to the end user.
    d. The user submits the signed request to the device which uses it to install the end user’s public key as the new Owner Public Key, thereby removing Crypto4A’s public key and transferring ownership of the device from Crypto4A to the end user.
    e. Additional information regarding this procedure can be found at https://support.crypto4a.com/QxOS/Management/ownership_changing.html.
5. Put the module into FIPS validated state. This involves:
    a. The owner creates an authorization request to set mode of operation.
    b. The owner must sign the request and submit the signature to the QASM. If the signature is validated, the QASM processes the request and sets the module in the FIPS validated state.

Note: After installation, the Operator shall run the Show Version service i.e. running “*skm hsm info*" command to confirm that the output QASM version and firmware version matches the information in Table 2 above and Operational flag is set to “FIPS”.  If any of the above steps are altered or skipped the product is not considered as a FIPS-validated module and the Operational flag will be set to “STANDARD” indicating non validated configuration

## 11.2 Administrator Guidance

During the manufacturing process, Crypto4A is designated as the Owner of the QASM. Upon receiving the QASM, ownership shall be transferred to the customer. To do so, the customer will generate an ECDSA key pair and provide Crypto4A with the public key. Crypto4A will submit a signature using its private key accepting the transfer of ownership. After the transfer is complete, the customer is now designated as the Owner of the QASM, and the Crypto4A's ownership is removed.

## 11.3 Non-Administrator Guidance

Users will be given access to Crypto4A's support portal to access comprehensive documentation related to the QASM. Users can interact with the QASM from one of the connected single board computers (SBCs) present on the QxEdge or QxHSM using one of the various tools provided. The main tool is referred to as "spa-key-man". To see all available commands, run *spa-key-man help*.

## 11.4 End of Life

The end-of-life procedure for the module entails disabling FIPS-enabled mode, which zeroizes the entire module. An operator authenticated into the Owner role submits a zeroization request to the possessor of the OWNER private key, who would then authorize it by signing the authorization request with the OWNER private key. The vendor then sends the file back to the operator, who submits the request to the cryptographic module. The module then zeroizes all non-default keys, effectively resetting the module back to factory condition.

# 12 Mitigation of Other Attacks

This module does not implement security mechanisms to mitigate other attacks.