

GEOMYS

Geomys LLC

Go Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Oct 29, 2026

Prepared by:

Lightship Security

Applus

www.lightshipsec.com

Table of Contents

Table of Contents 2

List of Tables 4

List of Figures 5

1 General 6

1.1 Overview 6

1.2 Security Levels 6

2 Cryptographic Module Specification 7

2.1 Description 7

2.2 Tested and Vendor Affirmed Module Version and Identification 8

2.3 Excluded Components 24

2.4 Modes of Operation 24

2.5 Algorithms 24

2.6 Security Function Implementations 36

2.7 Algorithm Specific Information 43

2.8 RBG and Entropy 45

2.9 Key Generation 47

2.10 Key Establishment 47

2.11 Industry Protocols 48

3 Cryptographic Module Interfaces 48

3.1 Ports and Interfaces 48

4 Roles, Services, and Authentication 49

4.1 Authentication Methods 49
4.2 Roles 49
4.3 Approved Services 50
4.4 Non-Approved Services 63
4.5 External Software/Firmware Loaded 64
5 Software/Firmware Security 64
5.1 Integrity Techniques 64
5.2 Initiate on Demand 64
5.3 Open-Source Parameters 65
6 Operational Environment 65
6.1 Operational Environment Type and Requirements 65
7 Physical Security 65
8 Non-Invasive Security 65
9 Sensitive Security Parameters Management 66
9.1 Storage Areas 66
9.2 SSP Input-Output Methods 66
9.3 SSP Zeroization Methods 67
9.4 SSPs 67
9.5 Transitions 79
10 Self-Tests 79
10.1 Pre-Operational Self-Tests 79
10.2 Conditional Self-Tests 80

10.3 Periodic Self-Test Information 84
10.4 Error States 84
10.5 Operator Initiation of Self-Tests 85
11 Life-Cycle Assurance 85
11.1 Installation, Initialization, and Startup Procedures 85
11.2 Administrator Guidance 86
11.3 Non-Administrator Guidance 86
12 Mitigation of Other Attacks 86

List of Tables

Table 1: Security Levels 7
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 10
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 16
Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 24
Table 5: Modes List and Description 24
Table 6: Approved Algorithms 34
Table 7: Vendor-Affirmed Algorithms 34
Table 8: Non-Approved, Not Allowed Algorithms 36
Table 9: Security Function Implementations 43
Table 10: Entropy Certificates 45
Table 11: Entropy Sources 47
Table 12: Ports and Interfaces 49

Table 13: Roles 49

Table 14: Approved Services 63

Table 15: Non-Approved Services 64

Table 16: Storage Areas 66

Table 17: SSP Input-Output Methods 66

Table 18: SSP Zeroization Methods 67

Table 19: SSP Table 1 73

Table 20: SSP Table 2 79

Table 21: Pre-Operational Self-Tests 79

Table 22: Conditional Self-Tests 84

Table 23: Error States 85

List of Figures

Figure 1: Block Diagram 8

1 General

1.1 Overview

This non-proprietary FIPS 140-3 Security Policy for the Go Cryptographic Module v1.26.0 describes how the module meets the security requirements specified in FIPS 140-3 (Federal Information Processing Standard 140-3) for an overall security level 1 module and outlines the security rules and operating procedures required to maintain compliance.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1

Section	Title	Security Level
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	1
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

A software library that provides cryptographic functionality to the Go standard library and other Go applications.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The module's cryptographic boundary (represented by the red dotted line in Figure 1) consists of the entire Go Cryptographic Module executable code.

Tested Operational Environment's Physical Perimeter (TOEPP):

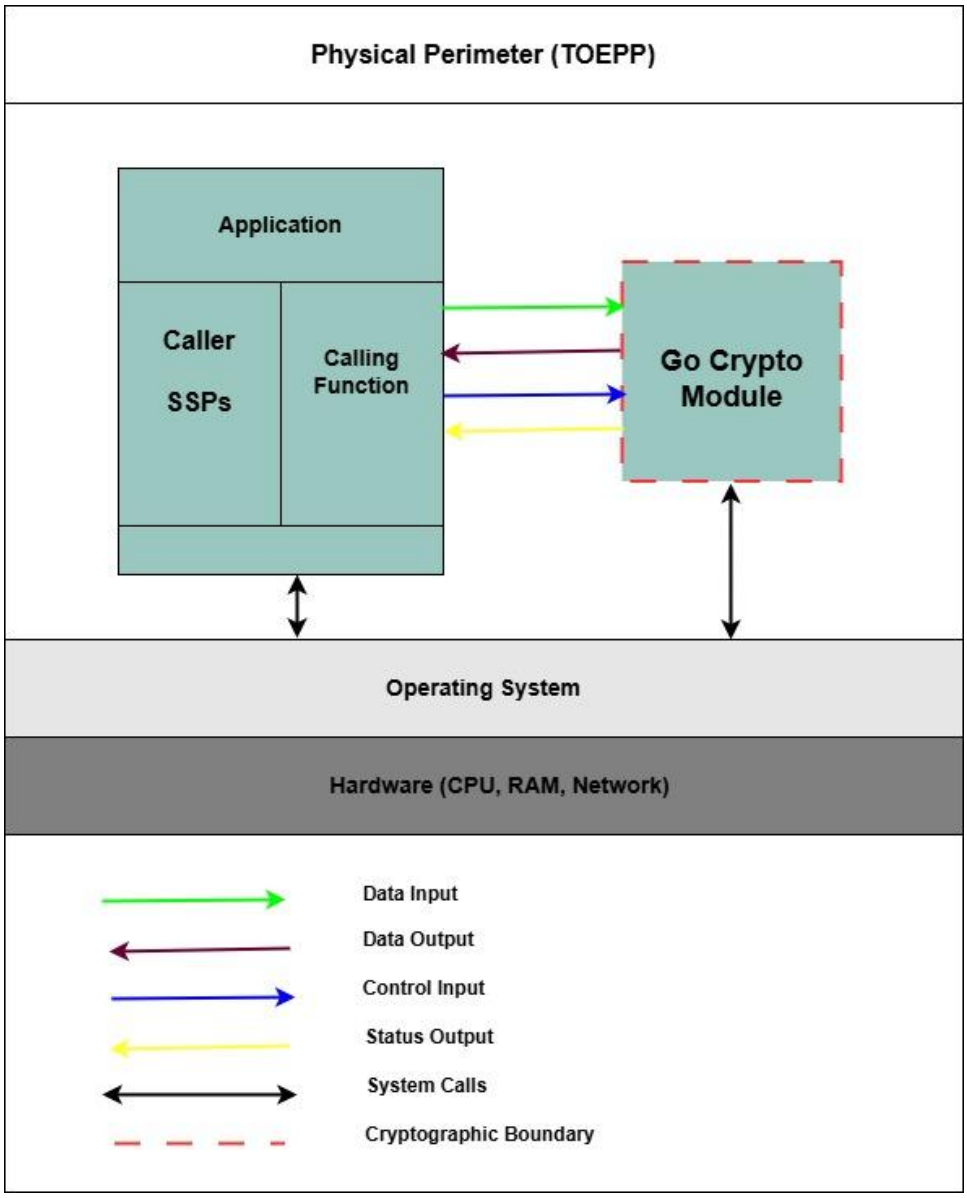


Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.o (darwin/arm64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (freebsd/amd64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/amd64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/arm)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/arm64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/mips64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/ppc64le)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (linux/s390x)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256
fips.o (windows/amd64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.o (windows/arm64)	v1.26.0	Binary file compiled from the source in the v1.26.0-cd900eea.zip file	HMAC-SHA2-256

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Red Hat Enterprise Linux 9	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	N/A	v1.26.0
Red Hat Enterprise Linux 9	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	N/A	v1.26.0
Red Hat Enterprise Linux 9	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	Yes	N/A	v1.26.0
Red Hat Enterprise Linux 9	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	No	N/A	v1.26.0
Red Hat Enterprise Linux 9	IBM z16 3931-A01	IBM Z System z16	Yes	PR/SM Driver Level 51 with Bundle Level H34	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Red Hat Enterprise Linux 9	IBM z16 3931-A01	IBM Z System z16	No	PR/SM Driver Level 51 with Bundle Level H34	v1.26.0
Red Hat Enterprise Linux 9	IBM 9080-HEX	IBM Power10	Yes	PowerVM FW1040.00 with VIOS 3.1.3.00	v1.26.0
Red Hat Enterprise Linux 9	IBM 9080-HEX	IBM Power10	No	PowerVM FW1040.00 with VIOS 3.1.3.00	v1.26.0
Alpine Linux 3.22 image	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	Podman 5 on Linux 6.12	v1.26.0
Alpine Linux 3.22 image	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	Podman 5 on Linux 6.12	v1.26.0
Alpine Linux 3.22 image	Supermicro H12SSW-AN6	AMD EPYC 7443P	Yes	Podman 5 on Linux 6.1	v1.26.0
Alpine Linux 3.22 image	Supermicro H12SSW-AN6	AMD EPYC 7443P	No	Podman 5 on Linux 6.1	v1.26.0
Alpine Linux 3.22 image	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	Yes	Podman 5 on Linux 6.12	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Alpine Linux 3.22 image	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	No	Podman 5 on Linux 6.12	v1.26.0
Amazon Linux 2023	Amazon EC2 r8g.metal-24xl	AWS Graviton4	Yes	N/A	v1.26.0
Amazon Linux 2023	Amazon EC2 r8g.metal-24xl	AWS Graviton4	No	N/A	v1.26.0
Amazon Linux 2023	Amazon EC2 c6i.metal	Intel Xeon Platinum 8375C	Yes	N/A	v1.26.0
Amazon Linux 2023	Amazon EC2 c6i.metal	Intel Xeon Platinum 8375C	No	N/A	v1.26.0
Oracle Linux 9	Oracle X9-2C	Intel Xeon Platinum 8358	Yes	N/A	v1.26.0
Oracle Linux 9	Oracle X9-2C	Intel Xeon Platinum 8358	No	N/A	v1.26.0
Oracle Linux 9	Oracle E6-2C	AMD EPYC 9J45	Yes	N/A	v1.26.0
Oracle Linux 9	Oracle E6-2C	AMD EPYC 9J45	No	N/A	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Oracle Linux 9	Oracle A1-2C	Ampere Altra Q80-30	Yes	N/A	v1.26.0
Oracle Linux 9	Oracle A1-2C	Ampere Altra Q80-30	No	N/A	v1.26.0
OpenWrt 21.02 with Linux 5.4	Ubiquiti EdgeRouter 4	Marvell OCTEON III CN7130	No	N/A	v1.26.0
OpenWrt 21.02 with Linux 5.4	Luxul ABR-4500	Broadcom BCM47094	No	N/A	v1.26.0
Google Prodimage with Linux 6.6.103	APIF-824	AMD EPYC 7B12	Yes	N/A	v1.26.0
Google Prodimage with Linux 6.6.103	APIF-824	AMD EPYC 7B12	No	N/A	v1.26.0
Google Prodimage with Linux 6.6.106	APIF-738	Intel Xeon Platinum 8273CL	Yes	N/A	v1.26.0
Google Prodimage with Linux 6.6.106	APIF-738	Intel Xeon Platinum 8273CL	No	N/A	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Google Prodimage with Linux 6.6.106	APIF-091	ARM Neoverse-N1	Yes	N/A	v1.26.0
Google Prodimage with Linux 6.6.106	APIF-091	ARM Neoverse-N1	No	N/A	v1.26.0
Apple macOS 15	MacBook Air (M2, 2022)	Apple M2	Yes	N/A	v1.26.0
Apple macOS 15	MacBook Air (M2, 2022)	Apple M2	No	N/A	v1.26.0
FreeBSD 14	Lenovo ThinkCentre M73	Intel Core i3-4130T	Yes	N/A	v1.26.0
FreeBSD 14	Lenovo ThinkCentre M73	Intel Core i3-4130T	No	N/A	v1.26.0
Microsoft Windows Server 2022	Lenovo ThinkCentre M73	Intel Core i3-4130T	Yes	N/A	v1.26.0
Microsoft Windows Server 2022	Lenovo ThinkCentre M73	Intel Core i3-4130T	No	N/A	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Microsoft Windows 11	Surface Laptop 7th Edition	Qualcomm Snapdragon X Plus	Yes	N/A	v1.26.0
Microsoft Windows 11	Surface Laptop 7th Edition	Qualcomm Snapdragon X Plus	No	N/A	v1.26.0
SUSE Linux Enterprise Server 16.0	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	N/A	v1.26.0
SUSE Linux Enterprise Server 16.0	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	N/A	v1.26.0
SUSE Linux Enterprise Server 16.0	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	Yes	N/A	v1.26.0
SUSE Linux Enterprise Server 16.0	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	No	N/A	v1.26.0
Red Hat Enterprise Linux 10	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	N/A	v1.26.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Red Hat Enterprise Linux 10	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	N/A	v1.26.0
Red Hat Enterprise Linux 10	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	Yes	N/A	v1.26.0
Red Hat Enterprise Linux 10	ASRock Rack ALTRAD8UD-1L2T	Ampere Altra Q64-22	No	N/A	v1.26.0
VMware Photon OS 5.0	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	ESXi 9.0	v1.26.0
VMware Photon OS 5.0	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	ESXi 9.0	v1.26.0
Ubuntu 24.04	Dell PowerEdge R660	Intel Xeon Silver 4410Y	Yes	ESXi 9.0	v1.26.0
Ubuntu 24.04	Dell PowerEdge R660	Intel Xeon Silver 4410Y	No	ESXi 9.0	v1.26.0

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform
Linux 3.10+	x86-64 architecture
Linux 3.10+	ARMv8 architecture
Linux 3.10+	ARMv9 architecture
Linux 3.10+	ARMv7 architecture
Linux 4.x	x86-64 architecture
Linux 4.x	ARMv8 architecture
Linux 4.x	ARMv9 architecture
Linux 4.x	ARMv7 architecture
Linux 5.x	x86-64 architecture
Linux 5.x	ARMv8 architecture
Linux 5.x	ARMv9 architecture
Linux 5.x	ARMv7 architecture
Linux 6.x	x86-64 architecture

Operating System	Hardware Platform
Linux 6.x	ARMv8 architecture
Linux 6.x	ARMv9 architecture
Linux 6.x	ARMv7 architecture
Linux 7.x	x86-64 architecture
Linux 7.x	ARMv8 architecture
Linux 7.x	ARMv9 architecture
Linux 7.x	ARMv7 architecture
Apple macOS 12	Apple M series
Apple macOS 13	Apple M series
Apple macOS 14	Apple M series
Apple macOS 15	Apple M series
Apple macOS 26	Apple M series
FreeBSD 12	x86-64 architecture

Operating System	Hardware Platform
FreeBSD 13	x86-64 architecture
FreeBSD 14	x86-64 architecture
FreeBSD 15	x86-64 architecture
Microsoft Windows 10	x86-64 architecture
Microsoft Windows 11	x86-64 architecture
Microsoft Windows 11	ARMv8 architecture
Microsoft Windows 11	ARMv9 architecture
Microsoft Windows Server 2016	x86-64 architecture
Microsoft Windows Server 2019	x86-64 architecture
Microsoft Windows Server 2022	x86-64 architecture
Microsoft Windows Server 2025	x86-64 architecture
Microsoft Windows Server 2025	ARMv8 architecture
Microsoft Windows Server 2025	ARMv9 architecture

Operating System	Hardware Platform
Red Hat Enterprise Linux 10	x86-64 architecture
Red Hat Enterprise Linux 10	ARMv8 architecture
Red Hat Enterprise Linux 10	ARMv9 architecture
Red Hat Enterprise Linux 10	IBM Z System z16
Red Hat Enterprise Linux 10	IBM Power10
Oracle Linux 7 with UEK6	Oracle X series servers
Oracle Linux 7 with UEK6	Oracle E series servers
Oracle Linux 7 with UEK6	Oracle A series servers
Oracle Linux 7 with UEK6	Marvell OCTEON TX2 CN93XX
Oracle Linux 7 with UEK6	AMD Pensando DSC2-2Q200
Oracle Linux 7 with UEK6	Marvell OCTEON III CN73XX
Oracle Linux 7 with UEK6	ARM Cortex-A7
Oracle Linux 7 with UEK6	ARM Cortex-A9

Operating System	Hardware Platform
Oracle Linux 8 with UEK7	Oracle X series servers
Oracle Linux 8 with UEK7	Oracle E series servers
Oracle Linux 8 with UEK7	Oracle A series servers
Oracle Linux 9 with UEK7	Oracle X series servers
Oracle Linux 9 with UEK7	Oracle E series servers
Oracle Linux 9 with UEK7	Oracle A series servers
Oracle Linux 9 with UEK7	Marvell OCTEON TX2 CN93XX
Oracle Linux 9 with UEK7	AMD Pensando DSC2-2Q200
Oracle Linux 9 with UEK7	Marvell OCTEON III CN73XX
Windows Server 2022	Dell XR4510c with Intel Xeon D-2776NT
Windows Server 2025	Dell XR4510c with Intel Xeon D-2776NT
Azure Local 22H2	Dell XR4510c with Intel Xeon D-2776NT
Red Hat Enterprise Linux 8.10 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT

Operating System	Hardware Platform
Red Hat Enterprise Linux 8.8 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Red Hat Enterprise Linux 8.5 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Red Hat Enterprise Linux 9.5 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Rocky Linux 8.5 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Ubuntu Linux 22.04 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Ubuntu Linux 24.04 on Hyper-V	Dell XR4510c with Intel Xeon D-2776NT
Amazon Linux 2023	AWS Graviton2
Amazon Linux 2023	AWS Graviton3
Amazon Linux 2023	x86-64 architecture
Amazon Linux 2023	ARMv8 architecture
Amazon Linux 2023	ARMv9 architecture
HPE Alletra Storage ArcusOS Release 10.7 (or later) on Debian GNU/Linux 12 (or later)	HPE Alletra Storage MP B10000

Operating System	Hardware Platform
Security Director Cloud 26.2.1 or later on Host OS Ubuntu 22.04 LTS and Container OS Ubuntu 24.04 LTS	VM (VMWare/KVM/Azure…)
Security Director On-Prem 26.2.1 or later on Host OS Ubuntu 20.04.6 LTS and Container OS Ubuntu 24.04 LTS	VM (VMWare/KVM/Azure…)
HPE Alletra Storage FleetOS Release 2.0.0 and up	HPE Alletra Storage MP X10000
HPE Mist Access Point Firmware 16.1 or later	AP47/AP47D/AP47E
HPE Mist Access Point Firmware 16.1 or later	AP36/AP36M, AP37, AP66/AP66D, AP17, AP27/AP27E, AP723H, AP774, AP775, AP775EX, AP777, AP777EX, AP779, AP779EX
HPE Mist Access Point Firmware 16.1 or later	AP714, AP715
Ubuntu 24.04	Generic ARM64 / AMD64 platform
Ubuntu 26.04	Generic ARM64 / AMD64 platform
Alpine Linux v3.22 (or later)	Generic ARM64 / AMD64 platform
Debian GNU/Linux 12 (Bookworm)	Generic ARM64 / AMD64 platform

Operating System	Hardware Platform
Debian GNU/Linux 13 (Trixie)	Generic ARM64 / AMD64 platform

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

There are no components excluded from the module's cryptographic boundary or the FIPS security requirements.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Types of Services	Status Indicator
Approved Mode	State in which the module is performing approved cryptographic services	Approved	ServiceIndicator() returns true
Non-Approved Mode	State in which the module is performing non-approved cryptographic services	Non-Approved	ServiceIndicator() returns false

Table 5: Modes List and Description

The module provides both approved and non-approved modes of operation. The approved mode of operation is entered automatically upon invocation of an approved service and exited automatically upon invocation of a non-approved service.

2.5 Algorithms

Approved Algorithms:

The table below lists the approved cryptographic algorithms of the module and implemented modes of operation of the algorithms.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A8028	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A8028	Direction - Generation, Verification Key Length - 128, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-524288 Increment 8	SP 800-38B
AES-CTR	A8028	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - Yes Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-GCM	A8028	Direction - Decrypt, Encrypt IV Generation - External, Internal Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 0-65536 Increment 8 AAD Length - AAD Length: 0-65536 Increment 8 IV Generation Mode - 8.2.2	SP 800-38D
Counter DRBG	A8028	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Additional Input - Additional Input: 384 Entropy Input - Entropy Input: 384 Nonce - Nonce: 0 Personalization String Length - Personalization String Length: 0 Returned Bits - 128	
cSHAKE-128	A8028	Message Length - Message Length: 0-65536 Increment 8 Output Length - Output Length: 16-65536 Increment 8 Hex Customization - No	SP 800-185
cSHAKE-256	A8028	Message Length - Message Length: 0-65536 Increment 8 Output Length - Output Length: 16-65536 Increment 8 Hex Customization - No	SP 800-185
Deterministic ECDSA SigGen (FIPS186-5)	A8028	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5
ECDSA KeyGen (FIPS186-5)	A8028	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A8028	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A8028	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384,	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
		SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	
ECDSA SigVer (FIPS186-5)	A8028	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5
EDDSA KeyGen	A8028	Curve - ED-25519	FIPS 186-5
EDDSA KeyVer	A8028	Curve - ED-25519	FIPS 186-5
EDDSA SigGen	A8028	Curve - ED-25519 Context Length - Context Length: 0-255 Increment 1 PreHash - Yes Pure - Yes	FIPS 186-5
EDDSA SigVer	A8028	Curve - ED-25519 PreHash - Yes Pure - Yes	FIPS 186-5
HMAC DRBG	A8028	Prediction Resistance - No Supports Reseed - No Mode - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Entropy Input - Entropy Input: 192, Entropy Input: 256 Nonce - Nonce: 128, Nonce: 96 Personalization String Length - Personalization String Length: 192, Personalization String Length: 256	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Additional Input - Additional Input: 0 Returned Bits - 224, 256, 384, 512	
HMAC-SHA2-224	A8028	MAC - MAC: 224 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A8028	MAC - MAC: 256 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A8028	MAC - MAC: 384 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A8028	MAC - MAC: 512 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A8028	MAC - MAC: 224 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A8028	MAC - MAC: 256 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A8028	MAC - MAC: 224 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A8028	MAC - MAC: 256 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A8028	MAC - MAC: 384 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA3-512	A8028	MAC - MAC: 512 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A8028	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder staticUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A8028	Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-65336 Increment 8 HMAC Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	SP 800-56C Rev. 2
KDA OneStepNoCounter SP800-56Cr2	A8028	Auxiliary Function Methods - Auxiliary Function Name - HMAC-SHA2-224 MAC Salting Methods - default, random Key Length - 224 Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Key Length - Key Length: 224-65336 Increment 8	SP 800-56C Rev. 2
KDF SP800-108	A8028	KDF Mode - Counter, Feedback MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-	SP 800-108 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 Supported Lengths - Supported Lengths: 256, Supported Lengths: 8-4096 Increment 8 Fixed Data Order - After Fixed Data, Before Fixed Data Counter Length - 16, 8 Supports Empty IV - No, Yes Custom Key In Length - 0 Requires Empty IV - Yes	
KDF SSH (CVL)	A8028	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KTS-IFC	A8028	IUT ID - C0FFEE Modulo - 2048, 3072, 4096 Key Generation Methods - rsakpg1-basic Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Associated Data Encoding - concatenation Key Length - 1024	SP 800-56B Rev. 2
ML-DSA KeyGen	A8028	Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87	FIPS 204
ML-DSA SigGen	A8028	Signature Interfaces - external, internal Pre Hash - pure	FIPS 204

Algorithm	CAVP Cert	Properties	Reference
		Deterministic - No, Yes External Mu - Yes Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87 Message Length - Message Length: 8-65536 Increment 8 Context Length - Context Length: 0-2040 Increment 8	
ML-DSA SigVer	A8028	Signature Interfaces - external, internal Pre Hash - pure External Mu - Yes Parameter Sets - ML-DSA-44, ML-DSA-65, ML-DSA-87 Message Length - Message Length: 8-65536 Increment 8 Context Length - Context Length: 0-2040 Increment 8	FIPS 204
ML-KEM EncapDecap	A8028	Parameter Sets - ML-KEM-1024, ML-KEM-768 Functions - Decapsulation, Encapsulation	FIPS 203
ML-KEM KeyGen	A8028	Parameter Sets - ML-KEM-1024, ML-KEM-768	FIPS 203
PBKDF	A8028	Iteration Count - Iteration Count: 1-10000 Increment 1 HMAC Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Password Length - Password Length: 8-64 Increment 1 Salt Length - Salt Length: 128-512 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-5)	A8028	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Fixed Public Exponent - 010001	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
		Info Generated By Server - Yes Private Key Format - standard Public Exponent Mode - fixed	
RSA SigGen (FIPS186-5)	A8028	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1	FIPS 186-5
RSA SigVer (FIPS186-5)	A8028	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1 Fixed Public Exponent - 010001 Public Exponent Mode - fixed	FIPS 186-5
SHA2-224	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-256	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-384	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-512	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512/224	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-512/256	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA3-224	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-256	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-384	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-512	A8028	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHAKE-128	A8028	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A8028	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
TLS v1.2 KDF RFC7627 (CVL)	A8028	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Key Block Length - Key Block Length: 1024	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A8028	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

The vendor affirms the approved implementation of the following algorithms.

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric and Asymmetric	Go Cryptographic Module	SP 800-133rev2 Section 4, Section 6.3

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

The module does not implement any non-approved or allowed algorithms in the approved mode of operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

The module does not implement any non-approved or allowed algorithms in the approved mode of operation, with no security claimed.

Non-Approved, Not Allowed Algorithms:

The table below lists the non-approved and not allowed cryptographic algorithms.

Name	Use and Function
AES ECB	Using Block.Encrypt, Block.Decrypt for encryption/decryption
AES GCM	Using arbitrary nonces for encryption
GHASH	MAC generation
Ed25519ctx	Signature generation/verification
HKDF	Using secrets shorter than 112 bits for key derivation; using a non-approved hash function for key derivation
HMAC	Using keys shorter than 112 bits for MAC generation; using a non-approved hash function for MAC generation
PBKDF	Using random salts shorter than 128 bits for key derivation; using a non-approved hash function for key derivation
TLS KDF	Using a non-approved hash function for key derivation
SSH KDF	Using a non-approved hash function for key derivation
RSA	Using key sizes smaller than 2048 bits or with odd bit length for key generation/verification, signature generation/verification, key encapsulation/decapsulation; using a non-approved hash function for signature generation/verification, key encapsulation/decapsulation; using public exponent smaller than 65537 for signature generation/verification, key encapsulation/decapsulation; primitive operation without padding; using non-approved DRBG for signature generation, key decapsulation; using primes of different lengths for signature generation, key

Name	Use and Function
	decapsulation; using PSS salt longer than hash output for signature generation/verification
ECDH	Using non-approved DRBG for shared secret computation
ECDSA	Using non-approved DRBG for key/signature generation; using a non-approved hash function for signature generation/verification

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Data Encryption	BC-Auth BC-UnAuth	Symmetric Encryption	Publication:FIPS 197	AES-CBC: (A8028) AES-CTR: (A8028) AES-GCM: (A8028)
Key Derivation Function	KAS-135KDF KAS-56CKDF PBKDF CKG	Key Derivation Function	Publication 1:SP80056Cr1 Publication 2:SP80056Cr2 Publication 3:SP800-108	KDA HKDF Sp800-56Cr1: (A8028) KDA OneStepNoCounter SP800-56Cr2: (A8028) KDF SP800-108: (A8028) KDF SSH: (A8028) PBKDF: (A8028) TLS v1.2 KDF RFC7627: (A8028) TLS v1.3 KDF: (A8028) CKG: ()

Name	Type	Description	Properties	Algorithms
				Key Type: Symmetric
Deterministic Random Bit Generation	DRBG CKG	Deterministic Random Bit Generation	Publication:SP800-90Ar1	Counter DRBG: (A8028) HMAC DRBG: (A8028) CKG: () Key Type: Symmetric
Digital Signature	DigSig-SigGen DigSig-SigVer	RSA, ECDSA SigGen/SigVer, EDDSA SigGen/SigVer	Publication 1:FIPS186-5	Deterministic ECDSA SigGen (FIPS186-5): (A8028) ECDSA SigGen (FIPS186-5): (A8028) ECDSA SigVer (FIPS186-5): (A8028) EDDSA SigGen: (A8028) EDDSA SigVer: (A8028) RSA SigGen (FIPS186-5): (A8028) RSA SigVer (FIPS186-5): (A8028)
Message Authentication	MAC	Generate or verify data integrity	Publication 1:SP800-38B Publication 2:FIPS 198-1	AES-CMAC: (A8028) HMAC-SHA2-224: (A8028)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-256: (A8028) HMAC-SHA2-384: (A8028) HMAC-SHA2-512: (A8028) HMAC-SHA2-512/224: (A8028) HMAC-SHA2-512/256: (A8028) HMAC-SHA3-224: (A8028) HMAC-SHA3-256: (A8028) HMAC-SHA3-384: (A8028) HMAC-SHA3-512: (A8028)
Key Agreement ECC	KAS-SSC	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module)	Security Strength:Key establishment methodology provides between 112 and 256 bits of encryption strength. Publication : NIST SP 800-56Arev3 IG : D.F - Scenario 2, Path (1)	KAS-ECC-SSC Sp800-56Ar3: (A8028)
Message digest	SHA	Hashing	Publication:FIPS 180-4	SHA2-224: (A8028) SHA2-256: (A8028) SHA2-384: (A8028) SHA2-512: (A8028) SHA2-512/224: (A8028) SHA2-512/256:

Name	Type	Description	Properties	Algorithms
				(A8028) SHA3-224: (A8028) SHA3-256: (A8028) SHA3-384: (A8028) SHA3-512: (A8028)
Key Generation	AsymKeyPair-KeyGen AsymKeyPair-KeyVer AsymKeyPair-PubKeyVal CKG	Non-PQC Asymmetric Key Generation	Publication 1:NIST SP 800-133rev2 - Section 5.1	ECDSA KeyGen (FIPS186-5): (A8028) ECDSA KeyVer (FIPS186-5): (A8028) EDDSA KeyGen: (A8028) EDDSA KeyVer: (A8028) RSA KeyGen (FIPS186-5): (A8028) CKG: () Key Type: Asymmetric
Key Encapsulation	AsymKeyPair-Decap AsymKeyPair-Encap	Key Encapsulation	Publication:SP800-56Br2	KTS-IFC: (A8028)
Data Decryption	BC-Auth BC-UnAuth	Symmetric Encryption	Publication:FIPS 197	AES-CBC: (A8028) AES-CTR: (A8028) AES-GCM: (A8028)

Name	Type	Description	Properties	Algorithms
Extendable Output Function	XOF	Output message with variable length	Publication:FIPS 202	cSHAKE-128: (A8028) cSHAKE-256: (A8028) SHAKE-128: (A8028) SHAKE-256: (A8028)
ML-KEM Encap	KEM-Encap	ML-KEM Encap	IG:D.S Scenario 1	ML-KEM EncapDecap: (A8028) Functions: Encapsulation SHAKE-256: (A8028) SHA3-512: (A8028) SHAKE-128: (A8028) SHA3-256: (A8028)
ML-KEM Decap	KEM-Decap	ML-KEM Decap	IG:D.S Scenario 1	ML-KEM EncapDecap: (A8028) Functions: Decapsulation SHAKE-256: (A8028) SHA3-512: (A8028) SHAKE-128: (A8028) SHA3-256: (A8028)

Name	Type	Description	Properties	Algorithms
ML-DSA Key Generation	AsymKeyPair-KeyGen CKG	ML-DSA Key Generation	Publication:FIPS204	ML-DSA KeyGen: (A8028) SHAKE-128: (A8028) SHAKE-256: (A8028) CKG: () Key Type: Asymmetric
ML-KEM Key Generation	AsymKeyPair-KeyGen CKG	ML-KEM Key Generation	Publication:FIPS203	ML-KEM KeyGen: (A8028) SHA3-256: (A8028) SHA3-512: (A8028) CKG: () Key Type: Asymmetric SHAKE-128: (A8028) SHAKE-256: (A8028)
ML-DSA External Hedged Sign	DigSig-SigGen	ML-DSA External Hedged Signature Generation	Publication:FIPS204	ML-DSA SigGen: (A8028) Signature Interface: External Deterministic: No SHAKE-256: (A8028)
ML-DSA External	DigSig-SigGen	ML-DSA External Deterministic	Publication:FIPS204	ML-DSA SigGen: (A8028) Signature Interface: External

Name	Type	Description	Properties	Algorithms
Deterministic Sign		Signature Generation		Deterministic: Yes SHAKE-256: (A8028)
ML-DSA External Sign (External Mu)	DigSig-SigGen	ML-DSA External Signature Generation (External Mu)	Publication:FIPS204	ML-DSA SigGen: (A8028) Signature Interface: External External Mu: Yes SHAKE-256: (A8028)
ML-DSA External Deterministic Sign (External Mu)	DigSig-SigGen	ML-DSA External Deterministic Signature Generation (External Mu)	Publication:FIPS204	ML-DSA SigGen: (A8028) Signature Interface: External Deterministic: Yes External Mu: Yes SHAKE-256: (A8028)
ML-DSA Internal Sign	DigSig-SigGen	ML-DSA Internal Signature Generation	Publication:FIPS204	ML-DSA SigGen: (A8028) Signature Interface: Internal SHAKE-256: (A8028)
ML-DSA External Verify	DigSig-SigVer	ML-DSA External Signature Verification	Publication:FIPS204	ML-DSA SigVer: (A8028) Signature Interface: External SHAKE-256: (A8028)

Name	Type	Description	Properties	Algorithms
ML-DSA External Verify (External Mu)	DigSig-SigVer	ML-DSA External Signature Verification (External Mu)	Publication:FIPS204	ML-DSA SigVer: (A8028) Signature Interface: External External Mu: Yes SHAKE-256: (A8028)
ML-DSA Internal Verify	DigSig-SigVer	ML-DSA Internal Signature Verification	Publication:FIPS204	ML-DSA SigVer: (A8028) Signature Interface: Internal SHAKE-256: (A8028)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

HMAC:

Per FIPS 140-3 IG C.M, key lengths below 112 bits are only allowed for legacy use (i.e. verification only) and the module does not allow them. However, HKDF uses the HMAC key for the salt, and PBKDF2 uses it for the password, which are allowed to be shorter.

PBKDF2:

Per FIPS 140-3 IG D.N, the module provides PBKDF API that accepts a password, salt, iteration count, and requested key length. The operator is responsible for selecting and applying one of the applicable options defined in NIST SP 800-132. The salt shall be at least 128 bits, the requested derived-key length shall be at least 112 bits, and the iteration count shall be at least 1,000. The module does not enforce the password length or character-set requirements; the operator is responsible for defining and enforcing them.

The probability of randomly guessing a password shall be determined by the operator after defining the password length and character set. The iteration count can be selected to be as large as practicable while maintaining an acceptable key-generation time. Keys derived from passwords using the PBKDF shall be used only in storage applications.

AES-GCM:

The module implements the following APIs to generate Initialization Vectors (IV) used in AES-GCM encryption:

- SealWithRandomNonce: 96 bits random nonces generated internally by Counter DRBG, complying with IG C.H Scenario 2;

- NewGCMForSSH: enforced construction of deterministic nonces as specified in RFC 5647, complying with IG C.H Scenario 1.d;

- NewGCMForTLS12: enforced construction of deterministic nonces as specified in RFC 5288, Section 3 and RFC 9325, Section 7.2.1, complying with C.H Scenario 1.a;

- NewGCMForTLS13: enforced construction of deterministic nonces as specified in RFC 8446, Section 5.3, complying with IG C.H Scenario 1.a;

- NewGCMWithCounterNonce: enforced construction of deterministic nonces where the first 32 bits must be an encoding of the module name and the last 64 bits must be a counter, complying with IG C.H Scenario 3;

- NewGCMForHPKE: enforced construction of deterministic nonces as specified in RFC 9180, Section 5.2, complying with IG C.H Scenario 5;

- NewGCMForQUIC: enforced construction of deterministic nonces as specified in RFC 9001, Section 5.3, complying with IG C.H Scenario 5;

- NewGCMWithXORCounterNonce: enforced construction of deterministic nonces where the first 32 bits must be an encoding of the module name and the last 64 bits must be a counter XOR'd with a fixed value, complying with IG C.H Scenario 3.

The IVs generated using NewGCMForSSH, NewGCMForTLS12,NewGCMForTLS13, NewGCMForHPKE, and NewGCMForQUIC should be strictly used in the context of the respective protocols. The counter portion of the IVs generated deterministically is strictly incremented and no more than 2^{64} - 1 encryptions may be performed in the same session. If the module's power is lost and then restored, the calling application is responsible for establishing new keys for AES-GCM encryption. No more than 2^{32} encryptions may be performed with SealWithRandomNonce with the same key.

KTS:

The module does not establish SSPs using an approved key transport scheme (KTS). However, it does offer approved authenticated algorithms that can be used by an external operator/application as part of an approved KTS.

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.

The shared secret key output by ML-KEM can be used as a key for symmetric cryptography. If further key derivation is needed, the shared secret key shall be used with the SP 800-108 KDFs or SP 800-56C KDFs specified in section 2.6 of this security policy.

[SP 800-227 RM3] When using an externally obtained ML-KEM static public encapsulation key, the operator shall ensure the other party's ownership of the key prior to invoking the ML-KEM encapsulation service. Such assurance may be obtained through a trusted third party (e.g., a certificate authority) or another mechanism appropriate for the deployment, consistent with RM3 and the guidance of Section 4.5 of SP 800-227.

[SP 800-227 RS6] If an application uses an ephemeral key pair, the application shall ensure that the key pair is used for only one execution of key-establishment via a KEM and shall be destroyed as soon as possible after its use.

[SP 800-227 RM2] When using ML-KEM, the calling application shall select an ML-KEM parameter set with a security strength appropriate for the intended application, in accordance with SP 800-227 Section 4.2 (RM2).

[SP 800-227 RM5] The calling application shall ensure that the key-establishment process over the communication channel provides an application-appropriate level of integrity, in accordance with SP 800-227 Section 4.2, RM5.

ML-DSA:

The module supports ML-DSA signature generation and verification using both the internal and external μ (Mu) interfaces. Per FIPS 204 for ML-DSA, the external Mu shall be provided by another (or the same) cryptographic module. The module that is used to compute the Mu externally must have its SHAKE256 implementation CAVP certified.

2.8 RBG and Entropy

Cert Number	Vendor Name
E318	Geomys LLC

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Go Entropy Source	Non-Physical	1. Alpine Linux 3.22 image on Podman 5 on Linux 6.1 running on AMD EPYC 7443P 2. Alpine Linux 3.22 image on Podman 5 on Linux 6.12 running on Ampere Altra Q64-22 3. Alpine Linux 3.22 image on Podman 5 on Linux 6.12 running on Intel Xeon Silver 4410Y 4. Amazon Linux 2023 running on AWS Graviton4 5. Amazon Linux 2023 running on Intel Xeon Platinum 8375C 6. Apple macOS 15 running on Apple M2 7. FreeBSD 14 running on Intel Core i3-4130T 8. Google Prodimage with Linux 6.6.103 running on AMD EPYC 7B12 9. Google Prodimage with Linux 6.6.106 running on ARM Neoverse-N1 10. Google Prodimage with Linux 6.6.106 running on Intel Xeon Platinum 8273CL 11. Microsoft Windows 11 running on Qualcomm Snapdragon X Plus 12. Microsoft Windows Server 2022 running on Intel Core i3-4130T 13. OpenWrt 21.02 with Linux 5.4 running on Broadcom BCM47094 14. OpenWrt 21.02 with Linux 5.4 running on Marvell OCTEON III CN7130 15. Oracle Linux 9 running on AMD EPYC 9J45 16. Oracle Linux 9 running on Ampere Altra Q80-30 17. Oracle Linux 9 running on Intel Xeon Platinum 8358 18. Red Hat Enterprise Linux 10 running on Ampere Altra Q64-22 19. Red Hat Enterprise Linux 10 running on Intel Xeon Silver 4410Y 20. Red Hat Enterprise Linux 9 on PR/SM Driver Level 51 with Bundle Level H34 running on IBM Z System z16 21. Red Hat	384 bits	384 bits	SHA2-384 (A7115)

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
		Enterprise Linux 9 running on Ampere Altra Q64-22 22. Red Hat Enterprise Linux 9 running on Intel Xeon Silver 4410Y 23. Red Hat Enterprise Linux 9 with PowerVM FW1040.00 with VIOS 3.1.3.00 running on IBM Power10 24. SUSE Linux Enterprise Server 16.0 running on Ampere Altra Q64-22 25. SUSE Linux Enterprise Server 16.0 running on Intel Xeon Silver 4410Y 26. Ubuntu 24.04 on ESXi 9.0 running on Intel Xeon Silver 4410Y 27. VMware Photon OS 5.0 on ESXi 9.0 running on Intel Xeon Silver 4410Y			

Table 11: Entropy Sources

The module's entropy source is located within the TOEPP, but outside the cryptographic boundary of the module. The module actively gets entropy based on its internal requirements but exercises no control over the quality of the obtained entropy.

2.9 Key Generation

When generating asymmetric keys, the module uses the output of CTR_DRBG to generate random numbers and keying material, per the guidance in NIST SP 800-133rev2, Section 5. ECDSA SigGen additionally generates the nonce according to FIPS 186-5, Appendix A.3.2 using a per-invocation instance of HMAC DRBG seeded with 256, 384, or 528 bits (for P-256, P-384, and P-521 respectively, per SP 800-90Ar1, Section 8.6.7) of entropy input from the global DRBG according to the IETF draft-irtf-cfrg-det-sigs-with-noise-03 document. This is analogous to Deterministic ECDSA specified in FIPS 186-5.

2.10 Key Establishment

The module implements the following approved key agreement method:

- KAS-ECC-SSC – NIST SP 800-56A Rev. 3 (FIPS 140-3 IG D.F, Scenario 2, Path (1))

While the module implements the protocol-specific KDFs in support of key agreement operations, the module only offers cryptographic services at the API level to calling applications and does not implement full key agreement within the module boundary.

The module implements the following key transport method:

- KTS-IFC – NIST SP 800-56B Rev. 2 (FIPS 140-3 IG D.G, Key Encapsulation/Decapsulation)

The module implements the following KEM method:

- ML-KEM – NIST FIPS 203

2.11 Industry Protocols

The module implements approved cryptography to support the following industry-specific protocols:

- SSH
- TLS v1.2
- TLS v1.3

However, the module only offers cryptographic services to calling applications and does not contain full implementations of industry protocols.

No parts of the SSH or TLS protocol, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	**Logical Interface(s)**	**Data That Passes**
N/A	Data Input	The module accepts data input through the input arguments of the API functions.
N/A	Data Output	The module produces data output through the parameters of the API functions.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Control Input	The module accepts control input through the input arguments of the API functions used to control the module.
N/A	Status Output	The module produces status output through the return values from function calls and error messages.

Table 12: Ports and Interfaces

As a software cryptographic toolkit, all of the module's interfaces are defined at the Software/Firmware Module Interface (SFMI). The logical interfaces are mapped to specific calls via a well-defined API, with which the operator interacts.

The data output interface is inhibited when the module is performing self-tests, performing zeroisation, or while in an error state.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement an authentication mechanism. The operator role of Crypto Officer is assumed implicitly.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	None

Table 13: Roles

The module only supports one role: Crypto Officer (CO). This role is assumed implicitly by the operator when performing an approved service.

4.3 Approved Services

The module performs the following approved services:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Data Encryption, Decryption	Encrypt or decrypt data	ServiceIndicator() returns true	Parameters, plaintext or ciphertext, key	Status, ciphertext or plaintext	Data Encryption Data Decryption	Crypto Officer - AES Key: W,E - AES GCM IV: W,E
Key Derivation Function	Perform key derivation using a key derivation function	ServiceIndicator() returns true	Parameters, key/password	Status, derived key	Key Derivation Function	Crypto Officer - KDF Derived Key: G,R - TLS Pre-Master Secret: G,R - TLS Master Secret: G,R - PBKDF Password: W,E - PBKDF Salt: W,E - PBKDF

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Derived Key: G,R
Deterministic Random Bit Generation	Generate random numbers with SP800-90A Rev 1	ServiceIndicator() returns true	Parameters, entropy	Status, random number	Deterministic Random Bit Generation	Crypto Officer - Entropy Input: W,E,Z - DRBG 'V' Value: W,E - DRBG 'K' Value: W,E - DRBG Seed: G,E,Z - HMAC DRBG Entropy: W,E - AES Key: G,R - AES GCM IV: G,R - AES GCM Key: G,R - AES

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						CMAC Key: G,R - ML-KEM Seed: G
Digital Signature	Generate or verify RSA, ECDSA, EdDSA digital signatures	ServiceIndicator() returns true	Parameters, RSA / ECDSA/ EdDSA keys, message	Status, digital signature	Digital Signature	Crypto Officer - RSA Public Key: W,E - RSA Private Key: W,E - ECDSA Public Key: W,E - ECDSA Private Key: W,E - EDDSA Public Key: W,E - EDDSA Private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: W,E - ML-DSA Private Key: E - ML-DSA Public Key: W,E
Message Authentication	Generate or verify data integrity	ServiceIndicator() returns true	Parameters, message, key	Status, message authentication code	Message Authentication	Crypto Officer - AES GCM Key: W,E - AES CMAC Key: W,E - HMAC Key: W,E
Key Agreement	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module)	ServiceIndicator() returns true	Parameters, ECDH keys	Status, shared secret	Key Agreement ECC	Crypto Officer - ECDH Peer Public Key: W,E - ECDH Private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: W,E - ECDH Shared Secret: G,R
Key Generation	Generate and verify an asymmetric keypair	ServiceIndicator() returns true	N/A	Status, keypair	Key Generation	Crypto Officer - RSA Public Key: G,R - RSA Private Key: G,R - ECDSA Public Key: G,R - ECDSA Private Key: G,R - EDDSA Private Key: G,R - EDDSA Public Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R - ECDH Public Key: G,R - ECDH Private Key: G,R - RSA Encap Key: G,R - RSA Decap Key: G,R
Key Encapsulation	Encapsulate/Decapsulate CSPs	ServiceIndicator() returns true	Parameters, plaintext or ciphertext key, transport key(s)	Status, plaintext or ciphertext key	Key Encapsulation	Crypto Officer - RSA Encap Key: W,E - RSA Decap Key: W,E
Message Digest	Generate a message digest	ServiceIndicator() returns true	Parameters, Message	Status, Digest of the message	Message digest	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Zeroize	Zeroize all SSPs	N/A	N/A	N/A	None	Crypto Officer - AES Key: Z - AES GCM IV: Z - AES GCM Key: Z - AES CMAC Key: Z - HMAC Key: Z - Entropy Input: Z - DRBG 'K' Value: Z - DRBG 'V' Value: Z - DRBG Seed: Z - RSA Public Key: Z - RSA Private Key: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						ECDSA Public Key: Z - ECDSA Private Key: Z - ECDH Public Key: Z - ECDH Private Key: Z - ECDH Shared Secret: Z - RSA Encap Key: Z - KDF Secret: Z - KDF Derived Key: Z - TLS Pre-Master Secret: Z - TLS Master Secret: Z - PBKDF

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Password: Z - PBKDF Salt: Z - PBKDF Derived Key: Z - EDDSA Private Key: Z - EDDSA Public Key: Z - ECDH Peer Public Key: Z - HMAC DRBG Entropy: Z - ML-DSA Private Key: Z - ML-DSA Public Key: Z - ML-KEM Decap Key: Z - ML-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KEM Encap Key: Z - ML-KEM SSK: Z - ML-KEM Seed: Z - ML-DSA Seed: Z
Self-tests	Perform the module self-tests	ServiceIndicator() returns true	N/A	Status	None	Crypto Officer
Show Status	Show status of the module	ServiceIndicator() returns true	N/A	Status	None	Crypto Officer
Show module's versioning information	Show module's versioning information	Module version	N/A	Status	None	Crypto Officer
Extendable Output Function	Output message with desired length	ServiceIndicator() returns true	Parameters, message, desired length	Status, message with desired length	Extendable Output Function	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
KEM Encap	Use a public key to generate a shared secret + ciphertext	ServiceIndicator() returns true	Parameters, KEM encap key	Status, shared secret, ciphertext	ML-KEM Encap	Crypto Officer - ML-KEM Encap Key: W,E - ML-KEM SSK: G,R
KEM Decap	Recover a shared secret using the private key and ciphertext	ServiceIndicator() returns true	Parameters, KEM decap key, ciphertext	Status, shared secret	ML-KEM Decap	Crypto Officer - ML-KEM Decap Key: E - ML-KEM SSK: G,R
ML-DSA Key Generation	Generate ML-DSA key pairs	ServiceIndicator() returns true	N/A	Status, key pair	ML-DSA Key Generation	Crypto Officer - ML-DSA Seed: G,E - ML-DSA Public Key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ML-DSA Private Key: G,R
ML-KEM Key Generation	Generate ML-KEM key pairs	ServiceIndicator() returns true	N/A	Status, key pair	ML-KEM Key Generation	Crypto Officer - ML-KEM Seed: G,E - ML-KEM Decap Key: G,R - ML-KEM Encap Key: G,R
ML-DSA Hedged Sign	ML-DSA Hedged Sign	ServiceIndicator() returns true	Parameters, private key, message, context	Status, signature	ML-DSA External Hedged Sign ML-DSA Internal Sign	Crypto Officer - ML-DSA Private Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
ML-DSA Deterministic Sign	ML-DSA Deterministic Sign	ServiceIndicator() returns true	Parameters, private key, message, context	Status, signature	ML-DSA External Deterministic Sign ML-DSA Internal Sign	Crypto Officer - ML-DSA Private Key: W,E
ML-DSA Hedged Sign (External Mu)	ML-DSA Hedged Sign (External Mu)	ServiceIndicator() returns true	Parameters, private key, mu	Status, signature	ML-DSA External Sign (External Mu) ML-DSA Internal Sign	Crypto Officer - ML-DSA Private Key: W,E
ML-DSA Deterministic Sign (External Mu)	ML-DSA Deterministic Sign (External Mu)	ServiceIndicator() returns true	Parameters, private key, mu	Status, signature	ML-DSA External Deterministic Sign (External Mu) ML-DSA Internal Sign	Crypto Officer - ML-DSA Private Key: W,E
ML-DSA Verify	ML-DSA Signature Verification	ServiceIndicator() returns true	Parameters, public key, signature,	Status	ML-DSA External Verify ML-DSA	Crypto Officer - ML-DSA Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			message, context		Internal Verify	Key: W,E
ML-DSA Verify (External Mu)	ML-DSA Signature Verification (External Mu)	ServiceIndicator() returns true	Parameters, public key, signature, mu	Status	ML-DSA External Verify (External Mu) ML-DSA Internal Verify	Crypto Officer - ML-DSA Public Key: W,E

Table 14: Approved Services

4.4 Non-Approved Services

The module provides the following non-approved services. Upon invocation of any non-approved services, `fips140.ServiceIndicator()` will return false until reset, and the module will exit the approved mode of operation.

Name	Description	Algorithms	Role
Data Encryption, Decryption	Encrypt or decrypt data	AES ECB	Crypto Officer
Data Encryption	Encrypt data	AES GCM	Crypto Officer
Message Authentication	Generate MAC	GHASH HMAC	Crypto Officer

Name	Description	Algorithms	Role
Digital Signature	Generate or verify digital signatures	Ed25519ctx RSA ECDSA	Crypto Officer
Key Derivation Function	Perform key derivation using a key derivation function	HKDF PBKDF TLS KDF SSH KDF	Crypto Officer
Key Agreement	Perform key agreement primitives on behalf of the calling process	ECDH	Crypto Officer

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support loading of external software.

5 Software/Firmware Security

5.1 Integrity Techniques

The module verifies the integrity of all software components within the cryptographic boundary using an HMAC-SHA2-256-based integrity technique. It applies HMAC-SHA2-256 to all components of the `fips.o` binary and compares it to the expected value embedded into the binary.

The module's pre-operational integrity check is performed automatically at module power-up.

5.2 Initiate on Demand

The module integrity check can be performed on demand by the module operator by rebooting the host platform or by reinitializing the module.

5.3 Open-Source Parameters

The source code of the module is part of the latest release of the open-source Go Programming Language project distributed under a BSD-style license.

The module can be built using the Go toolchain which is also part of the Go Programming Language distribution. The version of the Go toolchain used to build the module should be the same as the one of the Go Programming Language distribution which contains the `lib/fips140/v1.26.0-cd900eea.zip` file.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

Operating systems provide process/memory isolation to prevent unauthorized access to CSPs and uncontrolled modifications of SSPs while the module is in process space. The module does not spawn processes or persistently store SSPs.

7 Physical Security

The cryptographic module is a multi-chip standalone software module and does not include any physical components. Therefore, no physical security mechanisms or protections are implemented, and the section 7 requirements do not apply to the module.

8 Non-Invasive Security

The module does not claim any mitigations against non-invasive attacks. Additionally, there are no non-invasive attack mitigations outlined in Annex F of ISO/IEC 19790:2012.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	System Memory	Dynamic
External	Calling Application	Dynamic

Table 16: Storage Areas

The table above lists the SSP storage areas implemented by the module. The module does not implement persistent storage of SSPs, and only stores keys temporarily in RAM, within the process space of the module.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
SSP Input	External	RAM	Plaintext	Manual	Electronic	
SSP Output	RAM	External	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

The table above lists the SSP Input-Output methods implemented by the module. The module only inputs or outputs SSPs using automated, electronic means to and from within the module TOEPP.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Reboot Host	Reboot	Memory is zeroized upon reboot	Operator Initiated (Procedural)
Module Unload	Module Unload	Module unloaded from memory	Operator Initiated (Procedural)

Table 18: SSP Zeroization Methods

The module does not persistently store keys; therefore, all keys are held in the module's process space in volatile memory and are effectively zeroized upon reboot of the host platform.

9.4 SSPs

The module supports the keys and other SSPs listed in the table below:

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	AES Key (CBC, CTR, GCM)	128,192, 256 - 128,192, 256	Symmetric Key - CSP	Deterministic Random Bit Generation		Data Encryption Data Decryption
AES GCM IV	AES-GCM Initialization Vector	96 bits -	Key Component - CSP	Deterministic Random Bit Generation Key Derivation Function		Data Encryption Data Decryption

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES GCM Key	AES-GCM Key	128,192, 256 - 128,192, 256	Symmetric Key - CSP	Key Derivation Function		Data Encryption Data Decryption
AES CMAC Key	CMAC Key	128, 192, 256 - 128, 192, 256	Symmetric Key - CSP	Deterministic Random Bit Generation		Message Authentication
HMAC Key	Key used for HMAC Operations	>= 112 bits - >= 112 bits	HMAC Key - CSP	Deterministic Random Bit Generation		Message Authentication
Entropy Input	Externally generated entropy used to seed the DRBG	512 bits - 512 bits	Entropy - CSP			Deterministic Random Bit Generation
DRBG 'K' Value	CTR-DRBG Internal Value	256 bits - 256 bits	DRBG Internal State - CSP			Deterministic Random Bit Generation
DRBG 'V' Value	CTR-DRBG Internal Value	128 bits - 128 bits	DRBG Internal State - CSP			Deterministic Random Bit Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG Seed	DRBG Internal State Value	384 bits - 384 bits	DRBG Internal State - CSP			Deterministic Random Bit Generation
HMAC DRBG Entropy	Input to HMAC DRBG	256 - 528 bits - 256 - 528 bits	Entropy - CSP	Deterministic Random Bit Generation		Deterministic Random Bit Generation
RSA Public Key	Public Key used for RSA operations	>= 2048 bits - >= 112 bits	RSA Keypair - PSP	Deterministic Random Bit Generation		Digital Signature
RSA Private Key	Private Key used for RSA operations	>= 2048 bits - >= 112 bits	RSA Keypair - CSP	Deterministic Random Bit Generation		Digital Signature
ECDSA Public Key	Public Key used for EC operations	224 to 521 bits - 112 to 256 bits	EC Keypair - PSP	Deterministic Random Bit Generation		Digital Signature
ECDSA Private Key	Private Key used for EC operations	224 to 521 bits - 112 to 256 bits	EC Keypair - CSP	Deterministic Random Bit Generation		Digital Signature
ECDH Public Key	ECDH Public Key	224 to 521 bits - 112 to 256 bits	ECDH Keypair - PSP	Deterministic Random Bit Generation		Key Agreement ECC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDH Private Key	ECDH Private Key	224 to 521 bits - 112 to 256 bits	ECDH Keypair - CSP	Deterministic Random Bit Generation		Key Agreement ECC
ECDH Shared Secret	ECDH Shared Secret	224 to 521 bits - 112 to 256 bits	ECDH Shared Secret - CSP		Key Agreement ECC	Key Derivation Function
EDDSA Public Key	EDDSA Public Key	255 bits - 128 bits	EDDSA Keypair - PSP	Deterministic Random Bit Generation		Digital Signature
EDDSA Private Key	EDDSA Private Key	253 bits - 128 bits	EDDSA Keypair - CSP	Deterministic Random Bit Generation		Digital Signature
RSA Encap Key	RSA Encap Key	>= 2048 bits - >= 112 bits	Key Encapsulation - PSP	Key Generation		Key Encapsulation
KDF Secret	Secret used for KDF operations	>= 112 bits - >= 112 bits	Keying material - CSP		Key Agreement ECC	Key Derivation Function
KDF Derived Key	Key resulting from the module's SP	>= 112 bits - >= 112 bits	KDF Key - CSP	Key Derivation Function		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	800-135rev1 KDFs					
TLS Pre-Master Secret	Shared Secret used for TLS session establishment	112 to 256 bits - 112 to 256 bits	KDF Secret - CSP		Key Agreement ECC	Key Derivation Function
TLS Master Secret	Master Secret used for TLS session	112 to 256 bits - 112 to 256 bits	KDF Secret - CSP	Key Derivation Function		Data Encryption
PBKDF Password	Password used for password based key derivation	- - -	Password - CSP			Key Derivation Function
PBKDF Salt	Salt used for password based key derivation	>=128 bits - >=128 bits	Salt - Neither			Key Derivation Function
PBKDF Derived Key	Key resulting from the module's PBKDF	>= 112 bits - >= 112 bits	Symmetric Key - CSP	Key Derivation Function		Key Derivation Function

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDH Peer Public Key	ECDH Peer Public Key	224 to 521 bits - 112 - 256 bits	Asymmetric - PSP		Key Agreement ECC	KAS-ECC-SSC Sp800-56Ar3 (A8028)
ML-DSA Private Key	ML-DSA Private Key	20480, 32256, 39168 bits - 128, 192, 256 bits	Asymmetric - CSP	ML-DSA Key Generation		Digital Signature ML-DSA External Hedged Sign ML-DSA External Deterministic Sign ML-DSA External Sign (External Mu) ML-DSA External Deterministic Sign (External Mu)
ML-DSA Public Key	ML-DSA Public Key	10496, 15616, 20736 bits - 128, 192, 256 bits	Asymmetric - PSP	ML-DSA Key Generation		ML-DSA External Verify ML-DSA External Verify (External Mu) ML-DSA Internal Verify

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ML-KEM Decap Key	ML-KEM Decap Key	19200, 25344 bits - 192, 256 bits	Asymmetric - CSP	ML-KEM Key Generation		ML-KEM Decap
ML-KEM Encap Key	ML-KEM Encap Key	9472, 12544 bits - 192, 256 bits	Asymmetric - PSP	ML-KEM Key Generation		ML-KEM Encap
RSA Decap Key	RSA Decap Key	>= 2048 bits - >= 112 bits	Asymmetric - CSP	Key Generation		Key Encapsulation
ML-KEM SSK	Shared secret computed using KEM services	256 bits - 192 bits, 256 bits	Symmetric - CSP	ML-KEM Encap ML-KEM Decap		
ML-KEM Seed	Seed used to generate KEM key pairs	512 bits - 256 bits	Seed - CSP	Deterministic Random Bit Generation		ML-KEM Key Generation
ML-DSA Seed	Seed used to generate ML-DSA key pairs	256 bits - 256 bits	Seed - CSP	Deterministic Random Bit Generation		ML-DSA Key Generation

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
AES GCM IV	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
AES GCM Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	AES GCM IV:Derived From
AES CMAC Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
HMAC Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
Entropy Input	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
DRBG 'K' Value	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	DRBG 'V' Value:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG 'V' Value	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	DRBG 'K' Value:Used With
DRBG Seed	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
HMAC DRBG Entropy	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
RSA Public Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	RSA Private Key:Paired With
RSA Private Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	RSA Public Key:Paired With
ECDSA Public Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ECDSA Private Key:Paired With
ECDSA Private Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ECDSA Public Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ECDH Public Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ECDH Private Key:Paired With
ECDH Private Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ECDH Public Key:Paired With
ECDH Shared Secret	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ECDH Public Key:Used With ECDH Private Key:Used With
EDDSA Public Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	EDDSA Private Key:Paired With
EDDSA Private Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	EDDSA Public Key:Paired With
RSA Encap Key	SSP Input SSP Output	RAM:Encrypted External:Encrypted	Until Reboot	Reboot Host Module Unload	RSA Decap Key:Paired With
KDF Secret	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KDF Derived Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	KDF Secret:Derived From
TLS Pre-Master Secret	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	TLS Master Secret:Used With
TLS Master Secret	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	TLS Pre-Master Secret:Derived From
PBKDF Password	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	PBKDF Salt:Used With
PBKDF Salt	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	PBKDF Password:Used With
PBKDF Derived Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	PBKDF Password:Derived From PBKDF Salt:Derived From
ECDH Peer Public Key	SSP Input SSP Output	RAM:Plaintext	Until Reboot	Reboot Host Module Unload	ECDH Private Key:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ML-DSA Private Key		RAM:Plaintext	Until Reboot	Reboot Host Module Unload	ML-DSA Public Key:Paired With
ML-DSA Public Key	SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ML-DSA Private Key:Paired With
ML-KEM Decap Key		RAM:Plaintext	Until Reboot	Reboot Host Module Unload	ML-KEM Encap Key:Paired With
ML-KEM Encap Key	SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	ML-KEM Decap Key:Paired With
RSA Decap Key	SSP Input SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	RSA Encap Key:Paired With
ML-KEM SSK	SSP Output	RAM:Plaintext External:Plaintext	Until Reboot	Reboot Host Module Unload	
ML-KEM Seed		RAM:Plaintext	Until Reboot	Reboot Host Module Unload	ML-KEM Decap Key:Derives ML-KEM Encap Key:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ML-DSA Seed		RAM:Plaintext	Until Reboot	Reboot Host Module Unload	ML-DSA Private Key:Derives ML-DSA Public Key:Derives

Table 20: SSP Table 2

9.5 Transitions

A security strength of 112 bits will not be acceptable beyond 2031 in accordance with applicable CMVP transition requirements.

10 Self-Tests

10.1 Pre-Operational Self-Tests

The module performs the following pre-operational self-test:

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A8028)	HMAC-SHA2-256	Integrity Test	SW/FW Integrity	status output	Verify the integrity of the module

Table 21: Pre-Operational Self-Tests

The pre-operational self-tests are always performed on module instantiation and the HMAC-SHA2-256 CAST is performed before the integrity test, and therefore before the first operational use of the algorithm. If the self-tests fail, the module terminates.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 PCT	P-256	PCT	PCT	status output	KAS-ECC-SSC P-256, ECDH public key computation	Private key generation
ECDSA KeyGen (FIPS186-5) (A8028)	P-256, SHA2-512	PCT	PCT	status output	ECDSA sign and verify PCT	ECDSA private key generation
EDDSA KeyGen (A8028)	Ed25519	PCT	PCT	status output	EDDSA sign and verify PCT	EDDSA private key generation
ML-KEM KeyGen (A8028)	ML-KEM-768, ML-KEM-1024	PCT	PCT	status output	ML-KEM encap and decap PCT	ML-KEM-768/ML-KEM-1024 encapsulation key generation
RSA KeyGen (FIPS186-5) (A8028)	2048	PCT	PCT	status output	RSA sign and verify PCT	RSA private key generation
KAS-ECC-SSC Sp800-56Ar3 (A8028)	P256	KAT	CAST	status output	KAS-ECC-SSC P-256, ECDH secret key computation	Secret key computation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigGen (FIPS186-5) (A8028)	P256, SHA2-512	KAT	CAST	status output	ECDSA P-256 SHA2-512 sign and verify	Signature generation and signature verification
Deterministic ECDSA SigGen (FIPS186-5) (A8028)	P256, SHA2-512	KAT	CAST	status output	DetECDSA P-256 SHA2-512 sign	DetECDSA/ECDSA sign with HMAC-DRBG
EDDSA SigGen (A8028)	Ed25519, HashEd25519	KAT	CAST	status output	Ed25519 sign and verify	Signature generation and signature verification
RSA SigGen (FIPS186-5) (A8028)	2048 bits	KAT	CAST	status output	RSASSA-PKCS-v1.5 2048-bit sign and verify	Signature generation and signature verification
AES-CBC Encrypt	Key size: 128 bits	KAT	CAST	status output	AES-CBC encrypt	First time calling AES-CBC or AES-CTR
AES-CBC Decrypt	Key size: 128 bits	KAT	CAST	status output	AES-CBC decrypt	First time calling AES-CBC or AES-CTR
KDF SP800-108 (A8028)	Counter KDF AES-CMAC	KAT	CAST	status output	AES-128 CMAC Counter KDF	First time calling AES-GCM or AES-

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	Key size: 128 bits					CMAC or Counter KDF
Counter DRBG (A8028)	Entropy input: 384 bits	KAT	CAST	status output	ctrDRBG AES-256 KAT	First time calling Counter DRBG
cSHAKE-128 (A8028)	Security strength: 128 bits	KAT	CAST	status output	cSHAKE128 KAT	First time calling SHA3
HMAC DRBG (A8028)	hash: SHA2-512	KAT	CAST	status output	HMAC_DRBG SHA2-512 (Init, Reseed, Generate)	First time calling HMAC-DRBG API
TLS v1.3 KDF (A8028)	Extract: SHA2-256, Expand: SHA2-256	KAT	CAST	status output	TLS 1.3 KDF SHA2-256 KAT	First time calling TLS 1.3
TLS v1.2 KDF RFC7627 (A8028)	hash: SHA2-256	KAT	CAST	status output	TLS 1.2 RFC 7627 KDF SHA2-256 KAT	First time calling TLS 1.2
SHA2-512 (A8028)	Output size: 512 bits	KAT	CAST	status output	SHA2-512 KAT	First time calling SHA2-512
SHA2-256 (A8028)	Output size: 256 bits	KAT	CAST	status output	SHA2-256 KAT	First time calling SHA2-256

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ML-KEM Encap	ML-KEM-768 Encap	KAT	CAST	status output	ML-KEM-768 Encap KAT	First time calling ML-KEM
HMAC-SHA2-256 (A8028)	HMAC-SHA2-256	KAT	CAST	status output	HMAC-SHA2-256	First self-test that runs before module integrity check
HKDF Extract and Expand	HKDF Extract and Expand	KAT	CAST	status output	HKDF Extract and Expand	First time calling KDA HKDF, KDA OneStepNoCounter, or Feedback KDF
PBKDF (A8028)	SHA2-256	KAT	CAST	status output	PBKDF2 KAT	First time calling PBKDF2
ML-KEM KeyGen KAT	ML-KEM-768	KAT	CAST	status output	ML-KEM-768 keyGen KAT	First time calling ML-KEM keyGen
ML-DSA KeyGen (A8028)	ML-DSA-44, ML-DSA-65, ML-DSA-87	PCT	PCT	successful key generation	ML-DSA keyGen PCT	upon ML-DSA key generation
ML-DSA keyGen KAT	ML-DSA-44	KAT	CAST	status output	ML-DSA keyGen KAT	First time calling ML-DSA keyGen
ML-DSA SigGen (A8028)	ML-DSA-44	KAT	CAST	status output	ML-DSA sigGen KAT, covering all applicable rejection	First time calling ML-DSA sigGen

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
					sampling loop paths	
ML-DSA SigVer (A8028)	ML-DSA-44	KAT	CAST	status output	ML-DSA sigVer KAT	First time calling ML-DSA sigVer
ML-KEM Decap (Non-rejection)	ML-KEM-768 Decap	KAT	CAST	status output	ML-KEM Decap non-rejection KAT	First time calling ML-KEM
ML-KEM Decap (Implicit rejection)	ML-KEM-768 Decap	KAT	CAST	status output	ML-KEM Decap implicit rejection KAT	First time calling ML-KEM

Table 22: Conditional Self-Tests

Conditional CASTs are performed at module initialization, after the module integrity test, and before the first operational use of the algorithms. Pairwise consistency tests are performed conditionally upon generation of an asymmetric keypair.

10.3 Periodic Self-Test Information

N/A. The module has been tested to security level 1 and is not subject to periodic self-testing requirements.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The module's error state.	POST, PCT or CAST failure	Reinitialize module	fatal error

Table 23: Error States

The module terminates after any self-test failure, therefore no further requests to the module are available. To recover the module from error states, the host platform must be power cycled, or the module be reinitialized. Power cycling or module reinitialization will cause the module to perform the pre-operational self-tests and transition to the approved mode of operation.

10.5 Operator Initiation of Self-Tests

The module operator can initiate the pre-operational and conditional self-tests by power cycling the host platform or reinitializing the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module `fips.o` can be obtained by building the source code in the `lib/fips140/v1.26.0-cd900eea.zip` archive, included in the latest Go source distribution available at https://go.dev/dl.

The archive can be verified by issuing the command

```
sha256sum lib/fips140/v1.26.0-cd900eea.zip
```

and ensuring the SHA2-256 digest matches

```
9d85129ac28dfceee1634777673d2f712557d16da31534e608dd36a507be2912
```

The set of files in the archive verified as specified above is the complete set of source files of the validated module, and it shall be used unmodified in the build process.

Once the zip file has been obtained and verified, and the latest version of Go for your distribution was obtained and installed (for bootstrap), the following steps (or equivalent) can be executed to compile it as part of an application (at `$SRCROOT`, with the Go source distribution at `$GOROOT`).

```
cd "$GOROOT/src"
./make.bash
cd "$SRCROOT"
GOFIPS140=v1.26.0 "$GOROOT/bin/go" build # or equivalent
```

Alternatively, the GOFIPS140 variable can be specified when building the toolchain:

```
cd "$GOROOT/src"
GOFIPS140=v1.26.0 ./make.bash
cd "$SRCROOT"
"$GOROOT/bin/go" build # or equivalent
```

(On Windows, `make.bat` is used to build the toolchain instead of `make.bash`.)

Once the application (at `$APPBIN`) is compiled, the status of the module can be verified with the following command (or equivalent).

```
"$GOROOT/bin/go" version -m "$APPBIN"
```

If the output includes a line matching the following, the module has been configured correctly and can be operated in approved mode.

```
build GOFIPS140=v1.26.0-cd900eea
```

11.2 Administrator Guidance

The usage of approved algorithms and security functions shall conform to the Algorithm Specific Information described in section 2.7 of this security policy.

11.3 Non-Administrator Guidance

The module does not support any non-administrator roles or services. No guidance is warranted or available.

12 Mitigation of Other Attacks

The module implements the following mitigations:

1. During RSA signature generation processes, countermeasures are implemented to protect against vulnerabilities related to the Chinese Remainder Theorem (CRT), such as those exploited by Lenstra's attack.

2. The ECDSA scheme implements mitigation against both RNG failures (like Deterministic ECDSA) and fault attacks (like the RSA CRT attack mitigation).