ST Engineering Urban Solutions Ltd.

IPC Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

Introduction
Federal Information Processing Standards Publication 140-3 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140-3 program. The NVLAP accredits independent testing labs to perform FIPS 140-3 testing; the CMVP validates modules meeting FIPS 140-3 validation.

Validated is the term given to a module that is documented and tested against the FIPS 140-3 criteria.

More information is available on the CMVP website at:
https://csrc.nist.gov/projects/cryptographic-module-validation-program.

About this Document
This document describes the non-proprietary Security Policy for the IPC[1] Cryptographic Module (hereafter referred to as "the Module") from ST Engineering Urban Solutions Ltd. It contains specifications of the security rules under which the Module operates, including the security rules derived from the requirements of the FIPS 140-3 standard.

The following table lists the level of validation for each area in FIPS 140-3:
Overall Security Rating of the module is level 1.

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
10	Self-tests	1
11	Life-cycle assurance	3
12	Mitigation of other attacks	1
	Overall Level	1

Table 1: Security Levels

1.3 Additional Information

The Section 7.7 Physical Security and Section 7.8 Non-Invasive Security from ISO 19790 do not apply to the module.

[1] Internet Protocol Controller

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The module is intended to execute within the IPC device and provide cryptographic services.

Module Type: Software

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The cryptographic boundary is as depicted in Figure 1. No components are excluded from the cryptographic boundary. The module supports an Approved mode and a non-Approved mode of operation. The module does not support a degraded mode.

Tested Operational Environment's Physical Perimeter (TOEPP):

The block diagram of the Module is depicted in Figure 1 (blue outlined). The Tested Operational Environment's Physical Perimeter (TOEPP) is the underlying host platform i.e. IPC device on which it runs. The operating environment of the module is modifiable since the platform does support modifications to it.

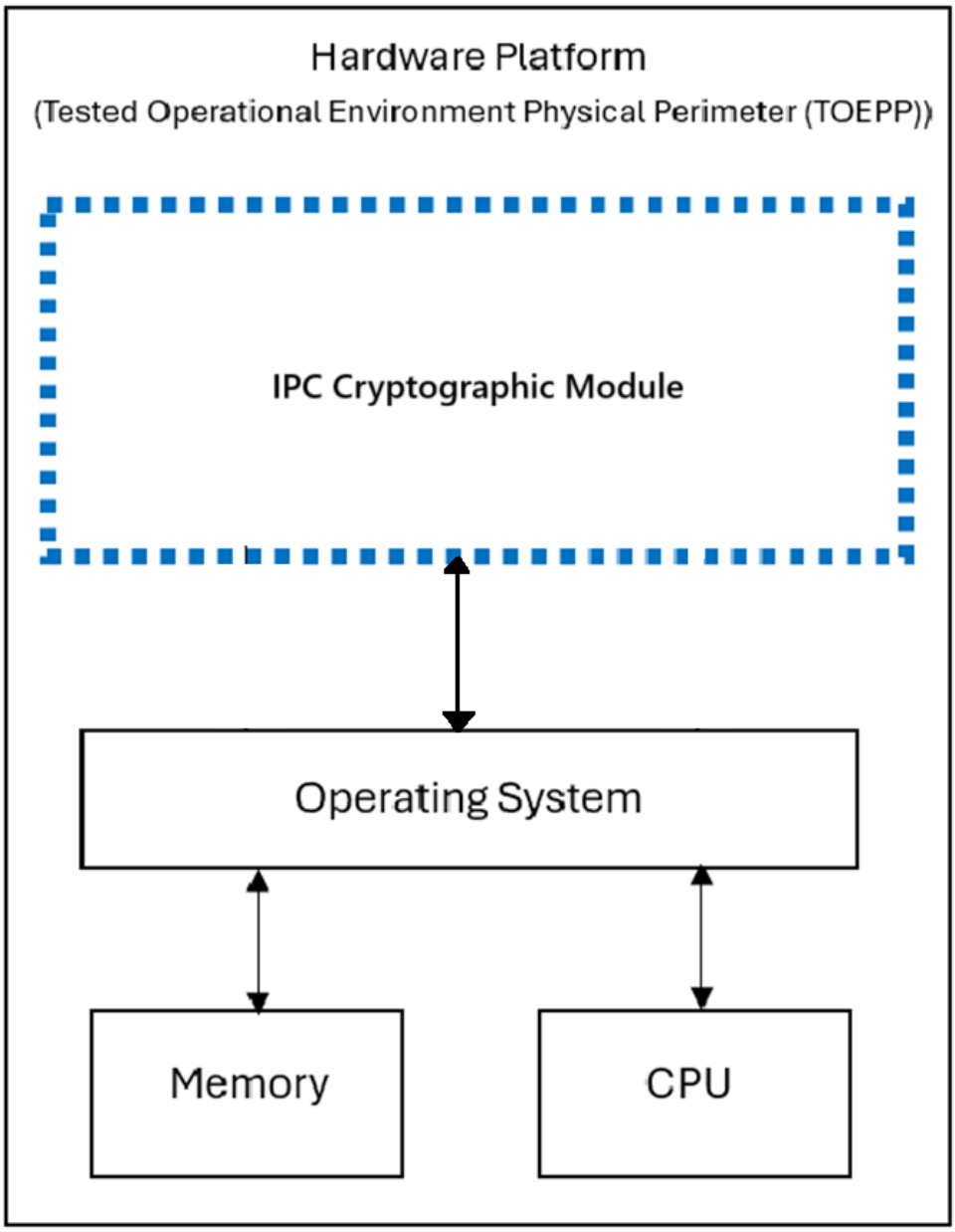

Figure 1: Block Diagram

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
IPC	V9FIPS.1.0	N/A	RSA mod 2048 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)
Windows 10 Enterprise LTSC Version 1809	IPC	Intel Core i5-8500T CPU @ 2.1Ghz	No	N/A	V9FIPS.1.0

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

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

No environments have been vendor affirmed.

2.3 Excluded Components

No components have been excluded.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	The module is initialized into the Approved mode of operation by default	Approved	"FIPS operation in progress" printed in bootlogs
non-Approved mode	The module transitions implicitly to the non-Approved mode upon usage of any Non-Approved Algorithms Not Allowed in the Approved Mode	Non-Approved	None

Table 4: Modes List and Description

The Module supports an Approved mode and a non-Approved mode of operation.

The following apply to the module:
1. The module does not support manual SSP entry.
2. The module inhibits data output during self-test execution, zeroisation, SSP generation and upon entry into the error state.

3. In the event of a self-test failure, all calls made to the module to request services from it are rejected by the module.

The Module is shipped with the Approved mode pre-enabled as noted in Section 11. No further configuration is required.

Mode Change Instructions and Status:

The module is in the Approved mode of operation provided the Approved algorithms and Non-Approved Algorithms Allowed in the Approved Mode are used. Usage of the non-Approved Algorithms Not Allowed in the Approved Mode causes the module to transition to the non-Approved mode.

Degraded Mode Description:

A degraded mode of operation is not supported by the module.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5153	-	SP 800-38A
AES-CBC-CS1	A5153	-	SP 800-38A
AES-CBC-CS2	A5153	-	SP 800-38A
AES-CBC-CS3	A5153	-	SP 800-38A
AES-CCM	A5153	-	SP 800-38C
AES-CFB1	A5153	-	SP 800-38A
AES-CFB128	A5153	-	SP 800-38A
AES-CFB8	A5153	-	SP 800-38A
AES-CMAC	A5153	-	SP 800-38B
AES-CTR	A5153	-	SP 800-38A
AES-ECB	A5153	-	SP 800-38A
AES-GCM	A5153	-	SP 800-38D
AES-GMAC	A5153	-	SP 800-38D
AES-KW	A5153	-	SP 800-38F
AES-KWP	A5153	-	SP 800-38F
AES-OFB	A5153	-	SP 800-38A
AES-XTS Testing Revision 2.0	A5153	-	SP 800-38E
Counter DRBG	A5153	-	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5153	-	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A5153	-	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5153	-	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5153	-	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
Hash DRBG	A5153	-	SP 800-90A Rev. 1
HMAC DRBG	A5153	-	SP 800-90A Rev. 1
HMAC-SHA-1	A5153	-	FIPS 198-1
HMAC-SHA2-224	A5153	-	FIPS 198-1
HMAC-SHA2-256	A5153	-	FIPS 198-1
HMAC-SHA2-384	A5153	-	FIPS 198-1
HMAC-SHA2-512	A5153	-	FIPS 198-1
HMAC-SHA2-512/224	A5153	-	FIPS 198-1
HMAC-SHA2-512/256	A5153	-	FIPS 198-1
HMAC-SHA3-224	A5153	-	FIPS 198-1
HMAC-SHA3-256	A5153	-	FIPS 198-1
HMAC-SHA3-384	A5153	-	FIPS 198-1
HMAC-SHA3-512	A5153	-	FIPS 198-1
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A5153	-	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A5153	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A5153	-	SP 800-56A Rev. 3
KAS-IFC-SSC	A5153	-	SP 800-56A Rev. 3
KDA HKDF SP800-56Cr2	A5153	-	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A5153	-	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A5153	-	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A5153	-	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A5153	-	SP 800-135 Rev. 1
KDF KMAC Sp800-108r1	A5153	-	SP 800-108 Rev. 1
KDF SP800-108	A5153	-	SP 800-108 Rev. 1
KDF SSH (CVL)	A5153	-	SP 800-135 Rev. 1
KMAC-128	A5153	-	SP 800-185
KMAC-256	A5153	-	SP 800-185
KTS-IFC	A5153	-	SP 800-56B Rev. 2
PBKDF	A5153	-	SP 800-132
RSA KeyGen (FIPS186-5)	A5153	-	FIPS 186-5
RSA SigGen (FIPS186-5)	A5153	-	FIPS 186-5
RSA Signature Primitive (CVL)	A5153	-	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-4)	A5153	-	FIPS 186-4
RSA SigVer (FIPS186-5)	A5153	-	FIPS 186-5
Safe Primes Key Generation	A5153	-	SP 800-56A Rev. 3
Safe Primes Key Verification	A5153	-	SP 800-56A Rev. 3
SHA-1	A5153	-	FIPS 180-4
SHA2-224	A5153	-	FIPS 180-4
SHA2-256	A5153	-	FIPS 180-4
SHA2-384	A5153	-	FIPS 180-4
SHA2-512	A5153	-	FIPS 180-4
SHA2-512/224	A5153	-	FIPS 180-4
SHA2-512/256	A5153	-	FIPS 180-4
SHA3-224	A5153	-	FIPS 202
SHA3-256	A5153	-	FIPS 202
SHA3-384	A5153	-	FIPS 202
SHA3-512	A5153	-	FIPS 202
SHAKE-128	A5153	-	FIPS 202
SHAKE-256	A5153	-	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A5153	-	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A5153	-	SP 800-135 Rev. 1

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG (6.3)	Key Type:Symmetric	N/A	NIST SP 800-133rev2, Section 6.3: Symmetric Keys Produced by Combining Multiple Keys and Other Data
CKG (4)	Key Type:Symmetric and Asymmetric	N/A	NIST SP800-133r2 Section 4: Using the Output of a Random Bit Generator; Section 5.1: Key Pairs for Digital Signature Schemes; Section 5.2: Key Pairs for Key Establishment; Section 6.1: Direct Generation of Symmetric Keys; Section 6.2: Derivation of Symmetric keys

Table 6: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

Name	Properties	Implementation	Reference
AES	Cert. A5153:key unwrapping per IG D.G	IPC	Symmetric key unwrapping per IG D.G Additional Comment 5
FIPS 186-4 RSA SigVer X9.31	Cert. A5153:signature verification	IPC	IG C.K

Table 7: Non-Approved, Allowed Algorithms

Non-Approved, Allowed Algorithms with No Security Claimed:

The module does not support any Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
X448	SSP Agreement
X25519	SSP Agreement
FIPS 186-5 ECDSA SigVer Component	Curve(s): P-192, P-224, P-256, P-384, P-521, B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, Function(s): SigVer
HMAC Generate	Key length(s): < 112 bits for MAC generation
HMAC DRBG/Hash DRBG	PRF(s): SHA3 (all sizes)
ED448	PRF: SHAKE256, Function(s): SigGen, SigVer
ED25519	PRF: SHA2-512, Function(s): SigGen, SigVer
TDES	Mode(s): CBC and ECB, Function(s): Encrypt, Decrypt
FIPS 186-4 DSA	Key size (strength): L = 1024, N = 160 (s < 112); L = 2048, N = 224 (s = 112); L = 2048, N = 256 (s = 112); L = 3072, N = 256 (s = 128); Function(s): KeyGen, SigGen, SigVer, PQGVer and PQGGen (SHA-1, SHA2 and SHA3 all sizes); SigVer and PQGVer disapproved per IG C.M 3.e
FIPS 186-2 RSA Signature	Modulus: > 1024 bits, Function(s): SigGen, SigVer (per IG C.M 3.e. for SigVer)
FIPS 186-2 RSA Generate Key	Modulus: >= 2048 bits, Function(s): KeyGen
KDA HKDF SP800-56Cr1	Key length(s): < 112 bits
KDA OneStep SP800-56Cr1	PRF(s): SHAKE128 and SHAKE256
KDF ANS 9.42	PRF(s): SHA-1, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256, KECCAK-KMAC128 and KECCAK-KMAC256
KDF ANS 9.63	PRF(s): SHA-1, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256, KECCAK-KMAC128 and KECCAK-KMAC256

Name	Use and Function
RSA PKCS1.5 (for KTS)	Usage of RSA PKCS1.5 Encapsulation/decapsulation in the context of SSP Transport (KTS)
RSA Signature Primitive	RSASP with modulus 3072, 4096 (since RSASP 2.0 is untested per CAVP Cert. #A5153)
FIPS 186-4 RSA KeyGen X9.31, FIPS 186-4 RSA SigGen X9.31	RSA KeyGen, SigGen per X9.31 per IG C.K
SHA-1 for SigVer	Usage of SHA-1 in the context of signature verification (per IG C.M 3.e)

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES Encrypt/Decrypt	BC-Auth BC-UnAuth	Encryption and decryption using AES modes	Key Length:128, 192 and 256 bits Key Length (XTS):128 and 256 bits	AES-CBC: (A5153) AES-CBC-CS1: (A5153) AES-CBC-CS2: (A5153) AES-CBC-CS3: (A5153) AES-CCM: (A5153) AES-CFB1: (A5153) AES-CFB128: (A5153) AES-CFB8: (A5153) AES-CMAC: (A5153) AES-CTR: (A5153) AES-ECB: (A5153) AES-GCM: (A5153) AES-GMAC: (A5153) AES-OFB: (A5153) AES-XTS Testing

Name	Type	Description	Properties	Algorithms
				Revision 2.0: (A5153)
AES Key Wrapping	KTS-Wrap	Key Wrapping	Key Length:128, 192 and 256 bits	AES-KW: (A5153) AES-KWP: (A5153)
SHS	SHA	Hashing		SHA-1: (A5153) SHA2-224: (A5153) SHA2-256: (A5153) SHA2-384: (A5153) SHA2-512: (A5153) SHA2-512/224: (A5153) SHA2-512/256: (A5153) SHA3-224: (A5153) SHA3-256: (A5153) SHA3-384: (A5153) SHA3-512: (A5153) SHAKE-128: (A5153) SHAKE-256: (A5153)
MAC	BC-Auth MAC	Message Authentication Code		HMAC-SHA-1: (A5153) HMAC-SHA2-224: (A5153) HMAC-SHA2-256: (A5153) HMAC-SHA2-384: (A5153) HMAC-SHA2-512: (A5153) HMAC-SHA2-512/224: (A5153) HMAC-SHA2-512/256: (A5153)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA3-224: (A5153) HMAC-SHA3-256: (A5153) HMAC-SHA3-384: (A5153) HMAC-SHA3-512: (A5153) AES-CMAC: (A5153) AES-GMAC: (A5153) KMAC-128: (A5153) KMAC-256: (A5153)
RSA SigGen/SigVer	DigSig-SigGen DigSig-SigVer	RSA SigGen and SigVer	Mode: PKCS 1.5 (SigGen):Modulus: 2048, 3072, 4096; Hash: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Mode: PKCSPSS (SigGen):Modulus: 2048, 3072, 4096; Hash: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Mode: ANSI X9.31 (SigVer only):Modulus: 1024, 2048, 3072, 4096; Hash: SHA2-256, SHA2-384, SHA2-512 Mode: PKCS 1.5 (SigVer):Modulus: 1024, 2048, 3072, 4096; Hash: SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Mode: PKCSPSS	RSA SigGen (FIPS186-5): (A5153) RSA SigVer (FIPS186-5): (A5153) RSA SigVer (FIPS186-4): (A5153)

Name	Type	Description	Properties	Algorithms
			(SigVer):Modulus: 1024, 2048, 3072, 4096; Hash: SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	
ECDSA SigGen/SigVer	DigSig-SigGen DigSig-SigVer	ECDSA SigGen and SigVer	SigGen:P-224, P-256, P-384, P-521, B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571; SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 SigVer :P-192, P-224, P-256, P-384, P-521, B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571; SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	ECDSA SigGen (FIPS186-5): (A5153) ECDSA SigVer (FIPS186-5): (A5153)
RSASP	DigSig-SigGen	RSA signature primitive		RSA Signature Primitive: (A5153)
Generate Key	AsymKeyPair-KeyGen AsymKeyPair-KeyVer CKG	Keypair generation		ECDSA KeyGen (FIPS186-5): (A5153) ECDSA KeyVer (FIPS186-5): (A5153) RSA KeyGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-5): (A5153) Safe Primes Key Generation: (A5153) Safe Primes Key Verification: (A5153) CKG (4): () Key Type: Symmetric and Asymmetric
Random Bit Generation	DRBG	Random Number Generation - Hash_DRBG, CTR_DRBG and HMAC_DRBG		Hash DRBG: (A5153) HMAC DRBG: (A5153) Counter DRBG: (A5153)
Derive	CKG KAS-135KDF KAS-56CKDF KBKDF PBKDF	Derive Keying Material		KDA HKDF SP800-56Cr2: (A5153) KDA OneStep SP800-56Cr2: (A5153) KDA TwoStep SP800-56Cr2: (A5153) KDF ANS 9.42: (A5153) KDF ANS 9.63: (A5153) KDF KMAC Sp800-108r1: (A5153) KDF SP800-108: (A5153) KDF SSH: (A5153) PBKDF: (A5153) TLS v1.2 KDF RFC7627: (A5153) TLS v1.3 KDF: (A5153) CKG (4): ()

Name	Type	Description	Properties	Algorithms
				Key Type: Symmetric and Asymmetric
KAS-1	KAS-SSC	Scheme: EphemeralUnified, KAS Role: Initiator, Responder	IG : IG D.F Scenario 2, path (1) Key confirmation:no Key derivation:no Caveat:Key establishment methodology provides between 112 and 256 bits of security strength	KAS-ECC-SSC Sp800-56Ar3: (A5153)
KAS-2	KAS-SSC	Scheme: dhEphem. KAS Role: Initiator, Responder	IG : IG D.F Scenario 2, path (1) Key confirmation:no Key derivation:no Caveat: Key establishment methodology provides between 112 and 200 bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A5153)
KAS-3	KAS-SSC	Scheme: KAS1, KAS2. KAS Role: Initiator, Responder	IG:IG D.F Scenario 1, path (1) Key confirmation :no Key derivation:no Caveat :Key establishment methodology provides between 112 and 200 bits of security strength	KAS-IFC-SSC: (A5153)
KTS-1	KTS-Wrap	Key Transport in compliance with [SP800- 38F] when approved using an Authenticated	Standard :SP 800-38F IG D.G:approved method from IG D.G Caveat :Key	AES-CCM: (A5153) AES-GCM: (A5153) AES-KW: (A5153)

Name	Type	Description	Properties	Algorithms
		AES mode (AES CCM; AES GCM; AES KW, KWP)	establishment methodology provides between 128 and 256 bits of security strength	AES-KWP: (A5153)
KTS-2	KTS-Wrap	Key Transport in compliance with [SP800- 38F] when approved AES (any mode) and approved HMAC are used in combination	Standard:SP 800-38F IG D.G:approved method from IG D.G Caveat :Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A5153) AES-CBC-CS1: (A5153) AES-CBC-CS2: (A5153) AES-CBC-CS3: (A5153) AES-CCM: (A5153) AES-CFB1: (A5153) AES-CFB128: (A5153) AES-CFB8: (A5153) AES-CMAC: (A5153) AES-CTR: (A5153) AES-ECB: (A5153) AES-GCM: (A5153) AES-GMAC: (A5153) AES-KW: (A5153) AES-KWP: (A5153) AES-OFB: (A5153) AES-XTS Testing Revision 2.0: (A5153) HMAC-SHA-1: (A5153) HMAC-SHA2-224: (A5153) HMAC-SHA2-256: (A5153)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-384: (A5153) HMAC-SHA2-512: (A5153) HMAC-SHA2-512/224: (A5153) HMAC-SHA2-512/256: (A5153) HMAC-SHA3-224: (A5153) HMAC-SHA3-256: (A5153) HMAC-SHA3-384: (A5153) HMAC-SHA3-512: (A5153)
KTS-3	KTS-Wrap	Key Transport in compliance with [SP800- 38F] when approved AES (any mode) and approved CMAC/GMAC are used in combination	Standard:SP 800-38F IG D.G:approved method from IG D.G Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A5153) AES-CBC-CS1: (A5153) AES-CBC-CS2: (A5153) AES-CBC-CS3: (A5153) AES-CCM: (A5153) AES-CFB1: (A5153) AES-CFB128: (A5153) AES-CFB8: (A5153) AES-CMAC: (A5153) AES-CTR: (A5153) AES-ECB: (A5153) AES-GCM: (A5153) AES-GMAC: (A5153) AES-KW: (A5153) AES-KWP: (A5153)

Name	Type	Description	Properties	Algorithms
				AES-OFB: (A5153) AES-XTS Testing Revision 2.0: (A5153)
KTS-4	KTS-Encap	Key Transport; Scheme: KTS-OAEP-basic (no key confirmation): RSA-OAEP, RSADP, RSAEP, Key Encapsulation, Key Unencapsulation Key Generation Methods: rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2- prime-factor	Standard:SP 800-56Brev2 IG D.G:approved method per IG D.G Key confirmation:no Caveat :Key establishment methodology provides between 112 and 176 bits of security strength	KTS-IFC: (A5153)
KAS ECC Component	KAS-SSC	KAS-ECC-SSC primitive (ECC CDH)		KAS-ECC CDH-Component SP800-56Ar3: (A5153)
Self-tests	BC-Auth BC-UnAuth DigSig-SigGen DigSig-SigVer DRBG KAS-135KDF KAS-56CKDF KAS-SSC KBKDF MAC PBKDF SHA XOF	All self-tests executed by the module at boot		AES-ECB: (A5153) AES-GCM: (A5153) Hash DRBG: (A5153) Counter DRBG: (A5153) HMAC DRBG: (A5153) ECDSA SigGen (FIPS186-5): (A5153) ECDSA SigVer (FIPS186-5): (A5153) RSA SigGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-5): (A5153) RSA SigVer (FIPS186-5): (A5153) HMAC-SHA2-256: (A5153) SHA-1: (A5153) SHA3-256: (A5153) KDF ANS 9.42: (A5153) KDF ANS 9.63: (A5153) KAS-ECC-SSC Sp800-56Ar3: (A5153) KAS-FFC-SSC Sp800-56Ar3: (A5153) KAS-IFC-SSC: (A5153) KDA OneStep SP800-56Cr2: (A5153) KDA TwoStep SP800-56Cr2: (A5153) KDA HKDF SP800-56Cr2: (A5153) KDF SSH: (A5153) PBKDF: (A5153) KDF SP800-108: (A5153) SHA2-512: (A5153) TLS v1.2 KDF RFC7627: (A5153) TLS v1.3 KDF: (A5153)
TLS all algorithms	AsymKeyPair-KeyGen AsymKeyPair-	All algorithms supported by the module for the		AES-GCM: (A5153) SHA2-384:

Name	Type	Description	Properties	Algorithms
	KeyVer BC-Auth CKG DigSig-SigGen DigSig-SigVer DRBG KAS-135KDF KTS-Wrap SHA	TLS 1.2 protocol/service		(A5153) RSA SigGen (FIPS186-5): (A5153) RSA SigVer (FIPS186-5): (A5153) ECDSA KeyGen (FIPS186-5): (A5153) Hash DRBG: (A5153) TLS v1.2 KDF RFC7627: (A5153) ECDSA SigGen (FIPS186-5): (A5153) ECDSA SigVer (FIPS186-5): (A5153) CKG (4): () Key Type: Symmetric and Asymmetric
Software Integrity Test	DigSig-SigVer	RSA mod 2048 bits SHA2-256 signature Verification		RSA SigVer (FIPS186-5): (A5153)
KTS-5	KTS-Wrap	Key wrapping in the context of the TLS 1.2 IETF protocol using an AES GCM 256-bit key	Standard:SP 800-38F IG D.G: approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-GCM: (A5153)
KAS-4	KAS-Full	Key agreement in the context of the TLS 1.2 IETF protocol; KAS-ECC-SSC P-384	IG : IG D.F Scenario 2 path (2) Key confirmation:no Key derivation:IG	KAS-ECC-SSC Sp800-56Ar3: (A5153) TLS v1.2 KDF RFC7627: (A5153)

Name	Type	Description	Properties	Algorithms
		used with KDF TLS 1.2	2.4.B SP 800-135rev1 CVL Caveat :Key establishment methodology provides 192 bits of security strength	
Symmetric Key Generation	CKG	Generation of symmetric keys		CKG (4): () CKG (6.3): ()

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

a. AES-GCM Usage
The AES GCM IV computation must comply with IG C.H and NIST SP 800-38D Scenario 1(a), tested per option (ii) under C.H TLS 1.2 protocol IV generation per RFC7627, Scenario 1(d) SSHv2 per RFC4252, RFC4253 and RFC5647 and Scenario 5 TLS 1.3 per RFC8446.

The Module does not implement the TLS 1.3 and SSH protocols itself, however, it provides the cryptographic functions required for implementing these protocols. The module does implement the TLS 1.2 protocol. AES GCM encryption is used in the context of the SSH and TLS protocol versions 1.2 and 1.3 and the IV computed shall only be used within the protocols. The module provides the primitives to support the AES GCM cipher suites per NIST SP800-52r1 Section 3.3.1. The module's implementation of AES-GCM is used together with an application that runs outside the module's cryptographic boundary in case of TLS 1.3 and SSH protocols. The application negotiates the protocol session's keys and the 32-bit nonce value of the IV.

When the IV exhausts the maximum number of possible values for a given session key (2^64 - 1), this results in failure in encryption and a handshake to establish a new encryption key will be required. It is the responsibility of the user of the module, i.e., the first party, client or server, to encounter this condition, to trigger this handshake in accordance with the TLS/SSH protocol.

The Module also supports internal IV generation using the module's approved DRBG. The IV is at least 96 bits in length per NIST SP800-38D Section 8.2.2. Per IG C.H Scenario 2 and NIST SP800-38D, the approved DRBG generates outputs such that the (key, IV) pair collision probability is less than 2^-32.

For all cases of IV generation, in the event that the module power is lost and restored the user must ensure that the AES GCM encryption/decryption keys are re-distributed/re-established in accordance with IG C.H Scenario 3. The module does not support persistent storage of SSPs.

The Module also supports importing of GCM IVs when an IV is not generated within the Module. In the approved mode, an IV must not be imported for encryption from outside the cryptographic boundary of the Module as this will result in a non-conformance. This is in accordance with IG 2.4.A: "*If the module operator (e.g., calling application) can do things outside of the module's control/visibility that can take an otherwise approved algorithm and use it in a non-approved way (e.g., use PBKDF and/or AES XTS outside of storage applications), the corresponding module service may still be considered approved (and if so, shall have an approved indicator per AS02.24) and the Security Policy shall clarify how to use the service in an approved manner (per ISO 19790 B.2.2 on Overall security design and the rules of operation).*"

b. AES-XTS Usage
Usage In accordance with NIST SP800-38E, the XTS-AES algorithm shall only be used for confidentiality on storage devices. The Module complies with IG C.I by explicitly checking that Key_1 ≠ Key_2 before using the keys in the XTS-AES algorithm to process data with them. The module implements CKG per NIST SP 800-133r2 Section 6.3.

c. Legacy Usage
The module supports the following implementations for legacy use/support per NIST SP 800-131Ar2:
    - FIPS 186-4/5 RSA (modulus 1024 bits), ECDSA (B-163, K-163 and P-192, curves) digital signature verification providing less than 112 bits of security strength. Legacy usage only. These legacy algorithms can only be used on data that was generated prior to the Legacy Date specified in IG C.M.

d. Component Validation List (CVL)
In accordance with IG 2.4.B, all tested components have been marked with the "CVL" notation in Table 5 and all vendor affirmed algorithms have been listed in Table 6. Also, per IG 2.4.B, the RSASP i.e. RSA SigGen (CVL) shall only be used within the context of a FIPS 186-5 signature generation.

e. PBKDF Usage
The module is compliant with IG D.N and NIST SP 800-132 Section 5.4 Option 1a. The iteration count values used range from 1 to 10000 per NIST SP 800-132 Section 5.2 whereby the iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The derived key must possess a minimum security strength of 112 bits. The module implements CKG per NIST SP 800-133r2 Section 6.2.2. In accordance with NIST SP 800-132 requirements, usage of the derived keys shall be restricted to storage applications alone.

The module supports a minimum 1-character long password. The ASCII system comprises of 94 printable characters (letters, digits, punctuation, and symbols). For a 1-character password/passphrase chosen from 94 printable ASCII characters, the total combinations are: 94^1. Thus, the probability of guessing the correct password/passphrase on a random attempt is: 1/94^1 ~ 0.01063.

The module being a software module does not restrict the usage of a password/string used as the password and input to the PBKDF. The onus is on the calling application to provide a password of an appropriate length based on the intended security strength (and size) of the key to be derived.

In accordance with NIST SP 800-132, passwords shorter than 10 characters are usually considered to be weak. There are many other properties that may render a password weak. For example, it is not advisable to use sequences of numbers or sequences of letters as passwords. Easily accessed personal information, such as the user's name, phone number, and date of birth, should not be used directly as a password.

Passphrases frequently consist solely of letters, but they make up for their lack of entropy by being much longer than passwords, typically 20 to 30 characters. Passphrases shorter than 20 characters are usually considered weak.

f. FIPS 202 Usage
Per IG C.C Resolution 2.a., each SHA-3 and SHAKE function has been tested and validated the module's operational environment.

g. RSA Usage
- Per IG C.E and IG C.F, the RSA SigGen and SigVer implementations have been tested for all implemented RSA modulus lengths where CAVP testing is available. The module supports generation of RSA keys with the following untested approved moduli/sizes: 4096 <nlen<= 16384. The module also supports the following untested, approved moduli for the RSA SigGen and SigVer: 4096 <nlen<= 16384.
- Per IG C.F Additional Comment 1.e:
The elliptic curves used in the key agreement scheme provide more than 112 bits of security as seen in the KAS entries per Table 10.
- Per IG C.F Additional Comment 2:
The KAS-ECC-SSC and KAS-FFC-SSC implementations support Diffie-Hellman based key agreement schemes.

h. TLS 1.2 KDF
Per IG D.Q, the module is compliant with RFC 7627 and is designed to enforce the usage of the extended master secret in the TLS 1.2 KDF.

i. NIST SP 800-108 KDF Usage
Per IG D.M, the SP 800-108 KDF is not used to generate asymmetric keys by the module. For keys provided by the calling application, the onus lies on the former to ensure that the keys have been generated using approved methods. The module supports CKG per NIST SP 800-133r2 Section 6.2.3.

2.8 RBG and Entropy

The Module complies with IG 9.3.A Scenario 2. b. and relies on the use of a NIST SP800-90B compliant entropy source outside the cryptographic boundary. The onus is on the calling

application to ensure the use of an NIST SP800-90B compliant entropy source and of sufficient entropy for the required security strength. The minimum number of bits of entropy, depending on the target security strength of generated SSPs is 128, 192 or 256 bits. If the Counter DRBG implementation without the derivation function enabled is used, ensure full entropy from the entropy source is provided. The following caveat applies to the module: No assurance of the minimum strength of generated SSPs (e.g., keys).

2.9 Key Generation

The module contains NIST SP 800-90Ar1 DRBGs and supports the NIST SP 800-133r2 (CKG) sections 4, 5.1, 5.2, 6.1, 6.2 and 6.3.

2.10 Key Establishment

The module supports key agreement and key transport in the context of the TLS protocol.

Apart from this, it also provides cryptographic primitives in support of key agreement and key transport where the onus is on the calling application to ensure that the primitives are used in the correct sequence. 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 (KTS-1, KTS-2, KTS-3 and KTS-4).

In addition to the TLS case, the following applies to the module for SSP agreement: The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS (KAS-1, KAS-2 and KAS-3).

Per IG D.F:

The module supports Key Agreement Schemes per NIST SP800-56Ar3 and IG D.F Scenario 2 (path 1) and NIST SP800-56Br2 and IG D.F Scenario 1 (path 1). The KAS-1, KAS-2, KAS-3 in the SFI Table 10 have been documented accordingly. The Approved Algorithm list includes the tested components (KAS-ECC-SSC, KAS-FFC-SSC and KAS-IFC-SSC) as individual entries.

The Module obtains the IG D.F required key agreement assurances:
NIST SP800-56Ar3 in accordance with Section 5.6.2.
NIST SP800-56Br2 in accordance with Section 6.4.

The module also supports key agreement in the context of the IETF TLS 1.2 protocol in accordance with IG D.F Scenario 2 (path 2) and the KAS-4 entry corresponds to the same.

Per IG D.G:

The module supports the Key Transport per NIST SP 800-56Br2 (RSA-OAEP) denoted by KTS-4 in the SFI Table 10. This notation is in accordance with the IG D.G Additional Comment 4: "*The FIPS 140-3 annotation details for the approved or allowed key transport schemes (KTS) can be found on SP 800-140B: CMVP Security Policy Requirements (see MIS Guidance "KTS").".*

The module also supports the following untested approved moduli for KTS-4: 6144 < nlen <=16384, where nlen denotes the modulus. The RSA modulus sizes and key generation method have been documented in the table as well. The module can also optionally be used in the context of IEFT protocols and provide key transport using any approved AES mode(s) and an approved MAC. The corresponding entries KTS-1, KTS-2 and KTS-3 in the SFI Table 10 have been documented accordingly. All KTS entries have been documented in accordance with Additional Comment 4 in the IG. Finally, KTS-5 corresponds to the key transport (wrapping) supported by the module in the context of the IETF TLS 1.2 protocol supported by it.

Per IG D.A and IG D.B:

The strengths of the established key have been documented in accordance with IG D.A Additional Comment 4. and per the Resolution in IG D.B.

2.11 Industry Protocols

The module supports cryptographic primitives used in the context of SSH, TLS 1.2 and TLS 1.3. It also supports the TLS 1.2 protocol itself. The module does not support the SSH and TLS 1.3 protocols and thus the following in accordance with Resolution #3 applies to the module:

No parts of the SSH and TLS 1.3 protocols, 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	Control Input	API input
N/A	Data Input	API parameters passed by calling applications for use in services
N/A	Status Output	API return code/status
N/A	Data Output	API parameters returned to calling applications as a result of service execution

Table 10: Ports and Interfaces

The module does not support control output and thus the Control Output interface is inapplicable.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The Module does not support authentication.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer (CO)	Role	Crypto Officer	None

Table 11: Roles

The module supports the Crypto Officer (CO) role alone, assumed implicitly by the calling application.

The module does not support a maintenance role, a bypass role or any unauthorized operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Module initialization	Module boot and initialization process	1	Ctx passed into the function	Return code 1 for success; 0 for failure	Random Bit Generation Software Integrity Test	Crypto Officer (CO) - Entropy Input: G,W,E,Z - State: G - Software Integrity Key - RSA: E
Show Status/Show Version	Show status; show version (version for fips.dll)	1	Ctx passed into the function fips_get_params	Status and versioning information	None	Crypto Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform Self-Tests	Execution of all self-tests	1	Reboot	Return code 1 for success; 0 for failure	Self-tests Software Integrity Test	Crypto Officer (CO)
Key Transport (Perform approved security functions)	Key encapsulation and unencapsulation	[KTS-IFC: RSA, 4, (2048, 3072, 4096, 6144, 8192)]	Encapsulation: SSP Transport Private Key; Decapsulation: SSP Transport Public Key	Key Transport Shared Secret	KTS-4	Crypto Officer (CO) - SSP Transport Private Key: E - SSP Transport Public key: E - Key Transport Shared Secret: R
Encrypt/Decrypt and Key Wrapping (Perform approved security functions)	Encrypt or decrypt data and key wrap	[AES-ECB: AES-128-ECB, AES-192-ECB, AES-256-ECB]; [AES-CBC: AES-128-CBC, AES-192-CBC, AES-256-	Symmetric Key and MAC Key (for wrapping)	Plaintext/ciphertext/wrapped key	AES Encrypt/Decrypt AES Key Wrapping KTS-1 KTS-2 KTS-3 Symmetric Key Generation	Crypto Officer (CO) - Symmetric Key: E - MAC Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		CBC]; [AES-CBC-CS: AES-128-CBC-CTS, AES-192-CBC-CTS, AES-256-CBC-CTS]; [AES-OFB: AES-128-OFB, AES-192-OFB, AES-256-OFB]; [AES-CFB1: AES-128-CFB1, AES-192-CFB1, AES-256-CFB1]; [AES-CFB8: AES-128-CFB8, AES-192-CFB8,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		AES-256-CFB8]; [AES-CFB128: AES-128-CFB, AES-192-CFB, AES-256-CFB]; [AES-CTR: AES-128-CTR, AES-192-CTR, AES-256-CTR]; [AES-CCM: AES-128-CCM, AES-192-CCM, AES-256-CCM]; [AES-GCM: AES-128-GCM, AES-192-GCM, AES-256-				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		GCM]; [AES-XTS: AES-128-XTS, AES-256-XTS]; [AES-KW, KWP: AES-128-WRAP, AES-256-WRAP]				
SSP Derivation (Perform approved security functions)	Derivation of keying material (DKM)	PBKDF: PBKDF2, (SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512)];	KAS Shared Secret	DKM	Derive	Crypto Officer (CO) - KAS Shared Secret: W,E - DKM: G,R - Key Transport Shared Secret: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		[TLS1-PRF, (SHA2-256, SHA2-384, SHA2-512)]; [TLS13-KDF, (SHA2-256, SHA2-384)]; [X963-KDF, (SHA2-224, SHA2-256, SHA2-384, SHA2-512)]; [X942KDF-ASNI, (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		SHA3-384, SHA3-512)]; [NIST SP 800-108r1 KDF KMAC: KBKDF, (KMAC-128, KMAC-256)]; [NIST SP 800-108r1 KDF: KBKDF, MAC: CMAC, Cipher: AES-128-CBC, AES-192-CBC, AES-256-CBC, MAC: HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512]; [KDF SSH: SSHKDF, (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512)]; [OneStep KDF: SSKDF				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		, (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		SHA3-224, SHA3-256, SHA3-384, SHA3-512, KMAC-128, KMAC-256)]; [TwoStep KDF: HKDF, MAC: HMAC, (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512]; [HKDF: HKDF, MAC: HMAC, (SHA1,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512]				
Key Agreement (Perform approved security functions)	Shared secret computation	[KAS-FFC-SSC: DHX]; [KAS-ECC-SSC: EC]	SSP Agreement Private FFC/ECC Key, SSP Agreement Public FFC/ECC Key	KAS Shared Secret	KAS-1 KAS-2 KAS-3 KAS ECC Component	Crypto Officer (CO) - SSP Agreement Private FFC/ECC Key: E - SSP Agreement Public FFC/ECC Key: E - KAS Shared Secret: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key Pair Generation (Perform approved security functions)	ECC/DH/RSA/ SafePrime key pair generation	[SafePrimes: DHX]; [RSA KeyGen: RSA, (2048, 3072, 4096)]; [ECDSA KeyGen: EC]	ECDSA: curve id. RSA: modulus	Key pair returned to caller	Generate Key	Crypto Officer (CO) - Private Key: G - Public Key: G
MAC Generation/Verification (Perform approved security functions)	Keyed hash generation/verification	[HMAC: HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384,	MAC Key	MAC value	MAC Symmetric Key Generation	Crypto Officer (CO) - MAC Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		HMAC-SHA3-512]; [CMAC]; [KMAC: KMAC-128, KMAC-256]; [GMAC: AES-128-GCM, AES-192-GCM, AES-256-GCM]				
Hash generation (Perform approved security functions)	Hashing	[SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128,	Message to be hashed	Hash value	SHS	Crypto Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		SHAKE-256]				
Random Bit Generation (Perform approved security functions)	Random bit generation using the DRBG	[Hash DRBG: HASH-DRBG, (SHA1, SHA2-256, SHA2-512)]; [HMAC-DRBG, (SHA1, SHA2-256, SHA2-512)]; [CTR-DRBG, (AES-128-CTR, AES-192-CTR, AES-256-CTR)]	DRBG State; DRBG Entropy Input	Random bits	Random Bit Generation	Crypto Officer (CO) - Entropy Input: E - Seed: E - State: E
Digital Signature Generation/Verification (Perform approved security functions)	RSA/ECDSA signature generation and verification	[RSA SigGen: RSA, (2048, 3072, 4096), (SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/22	Sign: SigGen Key; Verify: SigVer Key	Signature value for SigGen, 1 or 0 respectively for success or failure in case of SigVer	RSA SigGen/SigVer ECDSA SigGen/SigVer RSASP	Crypto Officer (CO) - SigGen Key: E - SigVer Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		4, SHA2-512/256)]; [RSA SigVer: RSA, (1024, 2048, 3072, 4096), (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256)]; [RSA Signature Primitive: RSA, 2048, hash algorithm: (null)]; [ECDSA SigGen: EC, (SHA2-224, SHA2-256, SHA2-384,				

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512)]; [ECDSA SigVer: EC, (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256)]; [ECDSA SigGen Component]: EC, hash: (null)]				
Perform zeroisation	* Zeroisation in the context of function calls * Restarting the host platform * TLS 1.2 Session Termination *	1	Location of SSP	Return code 1	None	Crypto Officer (CO) - SigGen Key: Z - SigVe

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	Module uninstantiation					r Key: Z - Private Key: Z - Public Key: Z - SSP Agreement Private FFC/ECC Key: Z - SSP Agreement Public FFC/ECC Key: Z - KAS Shared Secret: Z - DKM: Z - MAC Key: Z - SSP Transport Private Key: Z - SSP Transport Public key: Z - Key

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Transport Shared Secret: Z - Entropy Input: Z - Seed: Z - State: Z - Symmetric Key: Z - TLS Master Secret: Z - TLS Session Key: Z - KAS Public Key: Z - KAS Private Key: Z - ECDSA Public Key: Z - ECDSA Privat

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						e Key: Z - RSA Public Key: Z - RSA Private Key: Z - TLS Pre-Master Secret: Z - KAS Peer Public Key: Z - Software Integrity Key - RSA: Z
HTTPS communications with backend (Perform approved security functions)	TLS v1.2 protocol used	Successful completion of the service, i.e. successful TLS 1.2 session negotiation	TLS Peer Public Key (KAS Peer Public Key)	Packets transferred over TLS 1.2	TLS all algorithms KTS-5 KAS-4	Crypto Officer (CO) - TLS Master Secret: G,E,Z - TLS Pre-Master Secret: G,E,Z - TLS Session Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,E,Z - KAS Private Key: G,E,Z - KAS Public Key: G,R,E,Z - ECDSA Public Key: G,R,E,Z - ECDSA Private Key: G,E,Z - RSA Public Key: G,R,E,Z - RSA Private Key: G,E,Z - KAS Peer Public Key: W,E,Z

Table 12: Approved Services

The following indicate the type of access:

G = Generate: The service generates or derives the CSP/Public Key.
W = Write/Input: The service inputs the CSP/Public Key.
E = Execute: The Module executes using the CSP/Public Key.

R = Read/Output: The service outputs the CSP/Public Key. CSP are always protected with the approved KTS.

Z = Zeroise: The Module zeroizes the CSP/Public Key after usage. A zeroised CSP is not retrievable or reusable.

The module provides service indicators in accordance with the FIPS 140-3 IG 2.4.C example 3.

All CSPs are zeroised when they are no longer needed:

- Temporary CSPs are zeroised within the relevant function calls per service.
- The DRBG state is zeroised on Module instantiation
- The temporary underlying hash value generated as part of the RSA Signature Verification in the context of the integrity test performed, is zeroised prior to exiting the integrity test function.
- TLS 1.2 SSPs are zeroised upon TLS 1.2 session termination.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
SSP Agreement	KAS-SSC	X448 X25519	Crypto Officer (CO)
FIPS 186-5 ECDSA SigVer Component	Signature verification	FIPS 186-5 ECDSA SigVer Component	Crypto Officer (CO)
HMAC Generate	MAC generation with key length < 112 bits	HMAC Generate	Crypto Officer (CO)
HMAC DRBG/Hash DRBG	PRF(s): SHA3 (all sizes)	HMAC DRBG/Hash DRBG	Crypto Officer (CO)
TDES Encrypt/Decrypt	Encryption and decryption	TDES	Crypto Officer (CO)
Digital Signature Generation/Verification	Signature generation and verification	ED448 ED25519 FIPS 186-4 DSA FIPS 186-2 RSA Signature	Crypto Officer (CO)
FIPS 186-2 RSA Key Generation	RSA public/private key pair generation per FIPS 186-2	FIPS 186-2 RSA Generate Key	Crypto Officer (CO)
Derive	Key derivation	KDA HKDF SP800-56Cr1 KDA OneStep SP800-56Cr1 KDF ANS 9.42 KDF ANS 9.63	Crypto Officer (CO)

Name	Description	Algorithms	Role
SSP Transport	SSP transport using RSA PKCS1.5 padding	RSA PKCS1.5 (for KTS)	Crypto Officer (CO)
RSA Signature Primitive	Signature primitive function/signature generation with modulus 3072 and 4096	RSA Signature Primitive	Crypto Officer (CO)
FIPS 186-4 RSA X9.31 Key Generation and Signature Generation	Key generation and signature generation per ANS X9.31	FIPS 186-4 RSA KeyGen X9.31, FIPS 186-4 RSA SigGen X9.31	Crypto Officer (CO)
SHA-1 for Signature Verification	FIPS 186-4/5 RSA/ECDSA Signature Verification using SHA-1 (in accordance with IG C.M 3.e)	SHA-1 for SigVer	Crypto Officer (CO)

Table 13: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support loading software from an external source.

4.6 Bypass Actions and Status

The module does not support bypass.

4.7 Cryptographic Output Actions and Status

The module supports self-initiated cryptographic output over the TLS 1.2 IETF protocol. Two internal actions are performed, i.e. software flags are checked within the module prior to allowing output over TLS 1.2. The TLS 1.2 i.e. self-initiated cryptographic output capability is inherently active per the module's design. The Crypto Officer must only power on the underlying platform, i.e., IPC device. Successful negotiation of the TLS 1.2 session indicates that the self-initiated cryptographic output capability is active.

5 Software/Firmware Security

5.1 Integrity Techniques

The Module uses RSA 2048 SHA2-256 as the approved integrity technique. The pre-calculated value of the approved digital signature is included within the module. The integrity test covers the entirety of the module software. If the value calculated at boot for the approved digital signature does not match the pre-calculated, stored value, the test fails.

The RSA 2048 SHA2-256 CAST is performed prior to the software integrity test in accordance with the IG 10.2.A. The Module is provided in the executable form (.exe). The software integrity RSA mod 2048 public key used for signature verification is considered non-SSP and stored within the module.

5.2 Initiate on Demand

An operator of the module can perform the integrity test on demand by reloading the module. If the integrity test fails, module enters an error state. The module does not support loading any additional software from an external source.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The module is a Level 1 multi-chip standalone software module with a modifiable operational environment.

6.2 Configuration Settings and Restrictions

There are no restrictions on the operational environment of the module.

7 Physical Security

The Module is a software module thus the requirements per this section do not apply.

8 Non-Invasive Security

The Module is a software module thus the requirements per this section do not apply.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary, plaintext storage	Dynamic
Stored in the module binary	Persistent, plaintext storage	Static

Table 14: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input	Calling application	Module	Plaintext	Manual	Electronic	
API output	Module	Calling application	Plaintext	Manual	Electronic	
Stored at manufacture - 1	Manufacturer	Stored in the module binary	Plaintext	N/A	N/A	
Input during TLS 1.2 negotiation	TLS 1.2 peer/external endpoint	RAM	Plaintext	Automated	Electronic	
Output during TLS 1.2 negotiation	RAM	TLS 1.2 peer/external endpoint	Plaintext	Automated	Electronic	

Table 15: SSP Input-Output Methods

The module is compliant with IG 9.5.A MD/EE (CM Software to/from App via TOEPP Path) and with AD/EE in the context of the TLS 1.2 protocol.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroisation in the context of function calls	Temporary CSPs are zeroised within the relevant function calls per service	Automatic zeroisation per module's design in the context of each function called	Module initiated
Restarting the host platform	SSPs are stored temporarily in RAM	RAM is cleansed via reboot of the underlying host; no copies of SSPs are maintained/stored within the module itself	Operator initiated
TLS 1.2 Session Termination	SSPs zeroised upon TLS 1.2 session termination	TLS 1.2 SSPs are stored ephemerally until session termination	Module initiated

Zeroization Method	Description	Rationale	Operator Initiation
Module uninstantiation	DRBG state zeroisation	Un-instantiation of the module zeroises the DRBG state	Operator initiated

Table 16: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SigGen Key	Private key for signature generation	RSA: 2048, 3072 and 4096 bits ECDSA: B-233, K-233, P-224; B-283, K-283, P-256; B-409, K-409, P-384; B-571, K-571, P-521 - RSA: 112, 128 or 152 ECDSA: 112, 128, 192, 521	Private key - CSP			RSA SigGen/SigVer ECDSA SigGen/SigVer RSASP
SigVer Key	Public key for signature verification	RSA: 1024, 2048, 3072 and 4096 bits ECDSA: ECDSA: B-233, K-233, P-224; B-283, K-283, P-256; B-	Public key - PSP			RSA SigGen/SigVer ECDSA SigGen/SigVer

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		409, K-409, P-384; B-571, K-571, P-521 - RSA: 80, 112, 128 or 152 ECDSA: 112, 128, 192, 256				
Private Key	Private key requested by calling application (purpose unknown)	RSA: 2048, 3072, 4096 bits ECDSA: ECDSA: B-233, K-233, P-224; B-283, K-283, P-256; B-409, K-409, P-384; B-571, K-571, P-521 - RSA: 112, 128 or 152 ECDSA: 112, 128, 192, 256	Private key - CSP	Generate Key Random Bit Generation		
Public Key	Public key requested by calling application (purpose unknown)	RSA: 2048, 3072, 4096 bits ECDSA: ECDSA: B-233, K-233, P-224; B-283, K-	Public key - PSP	Generate Key Random Bit Generation		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		283, P-256; B-409, K-409, P-384; B-571, K-571, P-521 - RSA: 112, 128 or 152 ECDSA: 112, 128, 192, 256				
SSP Agreement Private FFC/ECC Key	Private key provided by the entity using the module for Diffie-Hellman shared secret generation	FFC: FB, FC, MODP2048, ffdhe2048, MODP3072, ffdhe3072, MODP4096, ffdhe4096, MODP6144, ffdhe6144, MODP8192, ffdhe8192 ECC: B-233, K-233, P-224, B-283, K-283, P-256, B-409, K-409, P-384, B-571, K-	Private key - CSP	Generate Key Random Bit Generation		KAS-1 KAS-2 KAS-3

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		571, P-521, IFC: k=2048, 3072, 4096, 6144, 8192 bits - FFC: between 112 and 200 ECC: 112, 128, 192, 256 IFC [SP800-56Br2]: 112, 128				
SSP Agreement Public FFC/ECC Key	Public key provided by the entity using the module for Diffie-Hellman shared secret generation	FFC: FB, FC, MODP2048, ffdhe2048, MODP3072, ffdhe3072, MODP4096, ffdhe4096, MODP6144, ffdhe6144, MODP8192, ffdhe8192 ECC: B-233, K-233, P-224, B-283, K-283, P-256, B-409, K-	Public key - PSP	Generate Key Random Bit Generation		KAS-1 KAS-2 KAS-3

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		409, P-384, B-571, K-571, P-521, IFC: k=2048, 3072, 4096, 6144, 8192 bits - FFC: between 112 and 200 ECC: 112, 128, 192, 256 IFC [SP800-56Br2]: 112, 128				
KAS Shared Secret	Shared secret computation (z)	FFC: FB, FC, MODP2048, ffdhe2048, MODP3072, ffdhe3072, MODP4096, ffdhe4096, MODP6144, ffdhe6144, MODP8192, ffdhe8192 ECC: B-233, K-233, P-224, B-283, K-	Shared secret - CSP		KAS-1 KAS-2 KAS-3	

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		283, P-256, B-409, K-409, P-384, B-571, K-571, P-521, IFC: k=2048, 3072, 4096, 6144, 8192 bits - FFC: between 112 and 200 ECC: 112, 128, 192, 256 IFC: 112, 128				
DKM	Key Derivation derived keying material	HMAC PRF: 160, 224, 256, 384, 512 - HMAC PRF: 160, 224, 256, 384, 512	Derived Keying Material - CSP		Derive	
MAC Key	Keyed Hash key	CMAC: 128, 192, 256 GMAC: 128, 192, 256 HMAC: 160, 256, 512. KMAC: 128, 256 - CMAC: 128, 192, 256 GMAC:	Symmetric key - CSP	Random Bit Generation Symmetric Key Generation		MAC KTS-2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		128, 192, 256 HMAC: 160, 256, 512. KMAC: 128, 256				
SSP Transport Private Key	Private key (KDK) used for [SP800-56Br2] RSA key transport	2048, 3072, 4096 and 6144 bits - 112, 128, 152, 176	Private key - CSP			KTS-4
SSP Transport Public key	Public key (KEK) used for [SP800-56Br2] RSA key transport	2048, 3072, 4096 and 6144 bits - 112, 128, 152, 176	Public key - PSP			KTS-4
Key Transport Shared Secret	The RSA key transport shared secret	2048, 3072, 4096 and 6144 bits - 112, 128, 152, 176	Shared secret - CSP			KTS-4
Entropy Input	Entropy input from an external source used for DRBG seeding	128 - 256 bits - 128 - 256 bits	Entropy input - CSP			Random Bit Generation
Seed	Seed generated from the entropy input for the DRBG	128 - 256 bits - 128 - 256 bits	DRBG seed - CSP			Random Bit Generation
State	DRBG state	Hash DRBG: 160, 224, 256, 384, 512 HMAC DRBG: 160, 224, 256, 384, 512. CTR DRBG:	DRBG state - CSP	Random Bit Generation		Random Bit Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		128, 192, 256 - Hash DRBG: 160, 224, 256, 384, 512 HMAC DRBG: 160, 224, 256, 384, 512. CTR DRBG: 128, 192, 256				
Symmetric Key	AES Encryption/Decryption/Key Wrapping Key	AES: 128, 192, 256 AES CCM: 128, 192, 256 AES GCM: 128, 192, 256 AES XTS: 128, 256. - AES: 128, 192, 256 AES CCM: 128, 192, 256 AES GCM: 128, 192, 256 AES XTS: 128, 256	Symmetric key - CSP	Random Bit Generation Symmetric Key Generation		AES Encrypt/Decrypt AES Key Wrapping KTS-1 KTS-2 KTS-3
TLS Master Secret	TLS 1.2 Master Secret derived from the pre-master secret	384 bits - 192 bits	Shared secret - CSP	Derive		TLS v1.2 KDF RFC7627 (A5153)
TLS Session Key	AES key used to encrypt the TLS session	256 bits - 256 bits	Symmetric key - PSP	Derive	KAS-4	KTS-5

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KAS Public Key	EC Diffie-Hellman i.e. KAS-ECC-SSC public key used in EC Diffie-Hellman Key Exchange for TLS 1.2	P-384 - 192 bits	Public key - PSP	Generate Key Random Bit Generation		KAS-ECC-SSC Sp800-56Ar3 (A5153)
KAS Private Key	EC Diffie-Hellman i.e. KAS-ECC-SSC private key used in EC Diffie-Hellman Key Exchange for TLS 1.2	P-384 - 192 bits	Private key - CSP	Generate Key Random Bit Generation		KAS-ECC-SSC Sp800-56Ar3 (A5153)
ECDSA Public Key	ECDSA public key used for TLS 1.2 authentication	P-384 - 192 bits	Public key - PSP	Generate Key Random Bit Generation		ECDSA SigVer (FIPS186-5) (A5153)
ECDSA Private Key	ECDSA private key used for TLS 1.2 authentication	P-384 - 192 bits	Private key - CSP	Generate Key Random Bit Generation		ECDSA SigGen (FIPS186-5) (A5153)
RSA Public Key	RSA public key used for TLS 1.2 authentication	RSA SigVer mod 2048 - 112 bits	Public key - PSP	Generate Key Random Bit Generation		RSA SigVer (FIPS186-5) (A5153)
RSA Private Key	RSA private key used for TLS 1.2 authentication	RSA SigGen mod 2048 - 112 bits	Private key - CSP	Generate Key Random Bit Generation		RSA SigGen (FIPS186-5) (A5153)
TLS Pre-Master Secret	TLS 1.2 pre-master secret computed (KAS-ECC-SSC)	384 bits - 192 bits	Shared secret - CSP		KAS-4	TLS v1.2 KDF RFC7627 (A5153)
KAS Peer Public Key	EC Diffie-Hellman i.e. KAS-ECC-SSC public key used in EC Diffie-Hellman Key Exchange for	P-384 - 192 bits	Public key - PSP			KAS-4

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	TLS 1.2 (TLS 1.2 peer key)					
Software Integrity Key - RSA	RSA key used to perform the Software Integrity Test	2048 bits - 112 bits	Public key - Neither		RSA SigVer (FIPS186-5) (A5153)	Software Integrity Test

Table 17: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SigGen Key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SigVer Key:Paired With
SigVer Key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SigGen Key:Paired With
Private Key	API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	Public Key:Paired With
Public Key	API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSP Agreement Private FFC/ECC Key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SSP Agreement Public FFC/ECC Key:Paired With
SSP Agreement Public FFC/ECC Key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SSP Agreement Private FFC/ECC Key:Paired With
KAS Shared Secret	API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SSP Agreement Private FFC/ECC Key:Established using SSP Agreement Public FFC/ECC Key:Established using
DKM	API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	
MAC Key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	
SSP Transport	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls	SSP Transport Public

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Private Key				Restarting the host platform Module uninstantiation	key:Paired With
SSP Transport Public key	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	SSP Transport Private Key:Paired With
Key Transport Shared Secret	API input API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	
Entropy Input	API input	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	Seed:Used to derive
Seed		RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform Module uninstantiation	Entropy Input:Derived From
State		RAM:Encrypted	Until power-cycling of the underlying host platform	Restarting the host platform Module uninstantiation	Seed:Derived From
Symmetric Key	API input API output	RAM:Encrypted	zeroised once no longer needed	Zeroisation in the context of function calls Restarting the host platform	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Module uninstantiation	
TLS Master Secret		RAM:Plaintext	zeroised once no longer needed	Zeroisation in the context of function calls	TLS Pre-Master Secret:Derived From
TLS Session Key		RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	TLS Master Secret:Derived From
KAS Public Key	Output during TLS 1.2 negotiation	RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	KAS Private Key:Paired With
KAS Private Key		RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	KAS Public Key:Paired With
ECDSA Public Key	Output during TLS 1.2 negotiation	RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	
ECDSA Private Key		RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	
RSA Public Key	Output during TLS 1.2 negotiation	RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	
RSA Private Key		RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	
TLS Pre-Master Secret		RAM:Plaintext	zeroised once no longer needed	Zeroisation in the context of function calls	TLS Master Secret:Used to derive
KAS Peer Public Key	Input during TLS 1.2 negotiation	RAM:Plaintext	zeroised once no longer needed	TLS 1.2 Session Termination	KAS Public Key:Used With
Software Integrity Key - RSA	Stored at manufacture - 1	Stored in the module binary:Plaintext	Until module uninstantiation is performed	Module uninstantiation	

Table 18: SSP Table 2

9.5 Transitions

Conformance to FIPS 186-5 is mandatory as of February 4, 2024. The module claims conformance to FIPS 186-4 as allowed per the FIPS 140-3 IG C.K Additional Comment #2.

Per the NIST SP 800-133Ar2/3 and the programmatic transitions defined by the CMVP, the following algorithm transitions apply to the module, and the algorithms have been designated allowed/non-approved accordingly in Section 2.5:

a. SHA-1 for SigVer (per IG C.M 3.e) and SHA-1 used for SigGen is a non-approved, not allowed algorithm. Usage of SHA-1 for all other SigVer is allowed for legacy use only until 2030. Thereafter, all usage of SHA-1 will be considered a non-approved, not allowed algorithm.
b. FIPS 186-2 RSA KeyGen and SigGen modes are non-approved, not allowed algorithms.
c. RSA-based key transport schemes that only use PKCS#1-v1.5 padding are non-approved, not allowed algorithms.
d. FIPS 186-4 DSA Key Gen, Sig Gen, or PQG Gen; FIPS 186-4 X9.31 RSA Key Gen, RSA Sig Gen are non-approved, not allowed algorithms.
e. Usage of FIPS 186-4 RSA SigVer X9.31 is allowed only for legacy use.
f. Triple-DES decryption is allowed for legacy use only.
g. Triple-DES encryption is non-approved, not allowed algorithm.
h. Key agreement schemes that are not compliant with any version of SP 800-56A (X448, X25519) are non-approved, not allowed algorithms.
i. Until January 1, 2031, the following algorithms will be considered deprecated:
    a. SHA-1, SHA-224 hash functions
    b. Hash_DRBG and HMAC_DRBG using SHA-1, SHA-224 hash functions
    c. Hash function and HMAC using SHA-1, SHA-224 hash functions
    d. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation
j. As of January 1, 2031, the following algorithms will be considered deprecated/disallowed (i.e. non-approved, not allowed)/legacy use:
    a. SHA-1, SHA2-224 hash functions (disallowed)
    b. Use of the 112-bit security strength for classical digital signature and key-establishment mechanisms (deprecated)
    c. Use of the 112-bit security strength for block ciphers (disallowed)
    d. Use of a security strength less than 128-bits but greater than 112 bits for ECDA KeyGen and RSA KeyGen (PKCS #1 v1.5 & PSS) (deprecated)
    e. Hash_DRBG and HMAC_DRBG using SHA-1, SHA-224 hash functions (disallowed)
    f. Hash function and HMAC using SHA-1, SHA-224 hash functions (legacy use)
    g. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation (disallowed)
    h. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Verification (legacy use)

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
RSA SigVer (FIPS186-5) (A5153)	Modulus: 2048 bits; Hash: SHA2-256	KAT	SW/FW Integrity	Verified OK	Verify

Table 19: Pre-Operational Self-Tests

The pre-operational self-tests can be run on demand by reloading the module.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A5153)	Key Length: 128 bits	KAT	CAST	1	Decrypt	On reloading the module
AES-GCM (A5153) - Encrypt - 256 bits	Key Length: 256 bits	KAT	CAST	1	Encrypt	On reloading the module
AES-GCM (A5153) - Decrypt - 256 bits	Key Length: 256 bits	KAT	CAST	1	Decrypt	On reloading the module
Counter DRBG (A5153)	AES CTR (128 bits) with derivation function	KAT	CAST	1	Generate, Reseed, Instantiate functions	On reloading the module
ECDSA SigGen (FIPS186-5) (A5153) - P-224	Curve: P-224; Hash: SHA2-512	KAT	CAST	1	Sign	On reloading the module
ECDSA SigVer (FIPS186-5) (A5153) - P-224	Curve: P-224; Hash: SHA2-512	KAT	CAST	1	Verify	On reloading the module
Hash DRBG (A5153)	PRF: SHA2-256	KAT	CAST	1	Generate, Reseed, Instantiate functions	On reloading the module
HMAC DRBG (A5153)	PRF: HMAC-SHA-1	KAT	CAST	1	Generate, Reseed, Instantiate functions	On reloading the module

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A5153)	PRF: SHA2-256	KAT	CAST	1	HMAC tag Generation	On reloading the module
KAS-ECC-SSC Sp800-56Ar3 (A5153)	Scheme: Ephemeral Unified, Curve: P-256	KAT	CAST	1	Key Agreement - Shared Secret Computation	On reloading the module
KAS-FFC-SSC Sp800-56Ar3 (A5153)	Scheme: dhEphem; Modulus: L = 2048 bits, N = 256 bit	KAT	CAST	1	Key Agreement - Shared Secret Computation	On reloading the module
KAS-IFC-SSC (A5153)	Schemes: Basic, CRT, Modulus: L = 2048 bits	KAT	CAST	1	Key Agreement - Shared Secret Computation	On reloading the module
KDF SP800-108 (A5153)	Mode: Counter, PRF: HMAC-SHA2-256	KAT	CAST	1	Counter Mode (HMAC-SHA2-256).	On reloading the module
KDA OneStep SP800-56Cr2 (A5153)	Auxiliary Function, H = SHA2-224	KAT	CAST	1	Key Derivation	On reloading the module
KDA TwoStep SP800-56Cr2 (A5153)	Auxiliary Function, H = HMAC-SHA2-256	KAT	CAST	1	Key Derivation	On reloading the module
KTS-IFC (A5153) - Basic	Schemes: Basic Modulus: L = 2048 bits	KAT	CAST	1	Encrypt	On reloading the module
KTS-IFC (A5153) - CRT	Schemes: Basic, CRT, Modulus: L = 2048 bits	KAT	CAST	1	Decrypt	On reloading the module
PBKDF (A5153)	Derivation of the Master Key (MK),	KAT	CAST	1	Key Derivation	On reloading the module

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	PRF: SHA2-256					
RSA SigGen (FIPS186-5) (A5153)	Scheme: PKCS#1, Modulus: L = 2048, Hash: SHA2-256	KAT	CAST	1	Sign	On reloading the module
RSA SigVer (FIPS186-5) (A5153)	Scheme: PKCS#1, Modulus: L = 2048, Hash: SHA2-256	KAT	CAST	1	Verify	On reloading the module
SHA-1 (A5153)	SHA-1	KAT	CAST	1	Hash	On reloading the module
SHA2-512 (A5153)	SHA2-512	KAT	CAST	1	Hash	On reloading the module
SHA3-256 (A5153)	SHA3-256	KAT	CAST	1	Hash	On reloading the module
KDF ANS 9.42 (A5153)	PRFs: AES KW (128 bits), SHA-1	KAT	CAST	1	Key Derivation	On reloading the module
KDF ANS 9.63 (A5153)	PRF: SHA2-256	KAT	CAST	1	Key Derivation	On reloading the module
KDF SSH (A5153)	PRF: SHA-1	KAT	CAST	1	Key Derivation	On reloading the module
TLS v1.2 KDF RFC7627 (A5153)	PRF: SHA2-256	KAT	CAST	1	Key Derivation	On reloading the module
TLS v1.3 KDF (A5153)	PRF: SHA2-256	KAT	CAST	1	Key Derivation	On reloading the module
RSA KeyGen (FIPS186-5) (A5153)	Performed on key generation	PCT	PCT	1	Key Generation	On generating keys for Key Transport (KTS IFC)/Key Agreement (KAS IFC)/Signature Generation/Signature Verification
ECDSA KeyGen (FIPS186-5) (A5153)	Performed on key generation	PCT	PCT	1	Key Generation	On generating keys for Key Agreement (KAS ECC)/Signature

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						Generation/Signature Verification
ECDSA SigGen (FIPS186-5) (A5153) - K-233	Curve: K-233; Hash: SHA2-512	KAT	CAST	1	Sign	On reloading the module
ECDSA SigVer (FIPS186-5) (A5153) - K-233	Curve: K-233; Hash: SHA2-512	KAT	CAST	1	Verify	On reloading the module
KAS-FFC-SSC Sp800-56Ar3 (A5153) - PCT	Performed post key generation	PCT	PCT	1	Key Generation	On generating keys for Key Agreement (KAS FFC)

Table 20: Conditional Self-Tests

The conditional cryptographic algorithm self-tests can be run on demand by reloading the module.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) (A5153)	KAT	SW/FW Integrity	On Demand	Manually by reloading the module

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A5153)	KAT	CAST	On Demand	Manually by reloading the module
AES-GCM (A5153) - Encrypt - 256 bits	KAT	CAST	On Demand	Manually by reloading the module
AES-GCM (A5153) - Decrypt - 256 bits	KAT	CAST	On Demand	Manually by reloading the module

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG (A5153)	KAT	CAST	On Demand	Manually by reloading the module
ECDSA SigGen (FIPS186-5) (A5153) - P-224	KAT	CAST	On Demand	Manually by reloading the module
ECDSA SigVer (FIPS186-5) (A5153) - P-224	KAT	CAST	On Demand	Manually by reloading the module
Hash DRBG (A5153)	KAT	CAST	On Demand	Manually by reloading the module
HMAC DRBG (A5153)	KAT	CAST	On Demand	Manually by reloading the module
HMAC-SHA2-256 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KAS-ECC-SSC Sp800-56Ar3 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KAS-FFC-SSC Sp800-56Ar3 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KAS-IFC-SSC (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDF SP800-108 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDA OneStep SP800-56Cr2 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDA TwoStep SP800-56Cr2 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KTS-IFC (A5153) - Basic	KAT	CAST	On Demand	Manually by reloading the module
KTS-IFC (A5153) - CRT	KAT	CAST	On Demand	Manually by reloading the module
PBKDF (A5153)	KAT	CAST	On Demand	Manually by reloading the module

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigGen (FIPS186-5) (A5153)	KAT	CAST	On Demand	Manually by reloading the module
RSA SigVer (FIPS186-5) (A5153)	KAT	CAST	On Demand	Manually by reloading the module
SHA-1 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
SHA2-512 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
SHA3-256 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDF ANS 9.42 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDF ANS 9.63 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
KDF SSH (A5153)	KAT	CAST	On Demand	Manually by reloading the module
TLS v1.2 KDF RFC7627 (A5153)	KAT	CAST	On Demand	Manually by reloading the module
TLS v1.3 KDF (A5153)	KAT	CAST	On Demand	Manually by reloading the module
RSA KeyGen (FIPS186-5) (A5153)	PCT	PCT	On Demand	On generation of keys
ECDSA KeyGen (FIPS186-5) (A5153)	PCT	PCT	On Demand	On generation of keys
ECDSA SigGen (FIPS186-5) (A5153) - K-233	KAT	CAST	On Demand	Manually by reloading the module
ECDSA SigVer (FIPS186-5) (A5153) - K-233	KAT	CAST	On Demand	Manually by reloading the module
KAS-FFC-SSC Sp800-56Ar3 (A5153) - PCT	PCT	PCT	On Demand	On generation of keys

Table 22: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard error	A failure in the pre-operational integrity test/one of the cryptographic algorithm self-tests will cause the module to return an error and enter the Hard error state	If the pre-operational software integrity test fails If one of the cryptographic algorithm's self-test (a CAST, specifically, a Known Answer Test (KAT)) were to fail	Reloading of the module	PROV_R_FIPS_MODULE_IN_ERROR_STATE and the string "signature not match"

Table 23: Error States

On instantiation, the Module performs the self-tests described in Table 22 and all CASTs. All KATs must complete successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the self-test failure state i.e. the Hard error state and returns the following indicator : “PROV_R_FIPS_MODULE_IN_ERROR_STATE” whereas if the integrity test fails, the module enters the Hard error state and returns the error string “signature not match” as well as “PROV_R_FIPS_MODULE_IN_ERROR_STATE”.

10.5 Operator Initiation of Self-Tests

The module can be reloaded on demand for running the Cryptographic Algorithm Self-tests (CASTs) as well as the pre-operational integrity test.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is shipped pre-installed in the Approved mode of operation with the IPC device. The operator must power on the appliance upon delivery to cause it to automatically execute the pre-operational integrity tests and CASTs on all algorithms. Once the self-tests have completed successfully, the module is ready for use. The operator can verify the software version (V9FIPS.1.0) returned via an API call to the module and the module identifier (the string “IPC” printed in the boot logs).

11.2 Administrator Guidance

No additional guidance applies for the operation of the module apart from that specified in Section 2 and other subsections under this section.

11.3 Non-Administrator Guidance

No additional guidance applies for the operation of the module apart from that specified in Section 2 and other subsections under this section.

11.4 Design and Rules

Azure is used as the Configuration Management System. The module's software is implemented using high-level language and designed to avoid use of code, parameters or symbols that are not necessary for the module's functionality and execution.

11.5 Maintenance Requirements

No maintenance requirements apply. The module's software is protected from tampering as it is delivered securely within the IPC device.

11.6 End of Life

The module can be uninstalled to end-of-life the module. The module can be securely sanitized by zeroising it.

12 Mitigation of Other Attacks

12.1 Attack List

The Module implements mitigations for some types of attacks using constant implementation and blinding.