IDEMIA

FIPS Applet on RookySE

FIPS 140-3 Non-Proprietary Security Policy

Version: 2.3

Revision Date: 10/28/2026

IDEMIA

Table of Contents

1 General 5
1.1 Overview 5
1.2 Security Levels 5
2 Cryptographic Module Specification 6
2.1 Description 6
2.2 Tested and Vendor Affirmed Module Version and Identification 8
2.3 Excluded Components 8
2.4 Modes of Operation 9
2.5 Algorithms 9
2.6 Security Function Implementations 14
2.7 Algorithm Specific Information 18
2.8 RBG and Entropy 18
2.9 Key Generation 19
2.10 Key Establishment 19
2.11 Industry Protocols 20
3 Cryptographic Module Interfaces 20
3.1 Ports and Interfaces 20
3.2 Additional Information 21
4 Roles, Services, and Authentication 21
4.1 Authentication Methods 21
4.2 Roles 26
4.3 Approved Services 28
4.4 Non-Approved Services 65
4.5 External Software/Firmware Loaded 66
4.6 Additional Information 66
5 Software/Firmware Security 66
5.1 Integrity Techniques 66
5.2 Initiate on Demand 67
5.3 Additional Information 67
6 Operational Environment 67
6.1 Operational Environment Type and Requirements 67
7 Physical Security 67
7.1 Mechanisms and Actions Required 67
7.2 EFP/EFT Information 68

7.3 Hardness Testing Temperature Ranges 68
8 Non-Invasive Security 69
9 Sensitive Security Parameters Management 69
9.1 Storage Areas 69
9.2 SSP Input-Output Methods 69
9.3 SSP Zeroization Methods 70
9.4 SSPs 71
9.5 Transitions 86
10 Self-Tests 87
10.1 Pre-Operational Self-Tests 87
10.2 Conditional Self-Tests 87
10.3 Periodic Self-Test Information 92
10.4 Error States 94
10.5 Operator Initiation of Self-Tests 95
11 Life-Cycle Assurance 95
11.1 Installation, Initialization, and Startup Procedures 95
11.2 Administrator Guidance 96
11.3 Non-Administrator Guidance 96
11.4 End of Life 96
11.5 Additional Information 96
12 Mitigation of Other Attacks 97
12.1 Attack List 97

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Hardware 8
Table 3: Modes List and Description 9
Table 4: Approved Algorithms 13
Table 5: Vendor-Affirmed Algorithms 13
Table 6: Non-Approved, Allowed Algorithms with No Security Claimed 14
Table 7: Security Function Implementations 18
Table 8: Entropy Certificates 18
Table 9: Entropy Sources 19
Table 10: Ports and Interfaces 21
Table 11: Authentication Methods 25
Table 12: Roles 26
Table 13: Approved Services 65
Table 14: Mechanisms and Actions Required 68
Table 15: EFP/EFT Information 68
Table 16: Hardness Testing Temperatures 69
Table 17: Storage Areas 69
Table 18: SSP Input-Output Methods 70
Table 19: SSP Zeroization Methods 71
Table 20: SSP Table 1 77
Table 21: SSP Table 2 86
Table 22: Pre-Operational Self-Tests 87
Table 23: Conditional Self-Tests 91
Table 24: Pre-Operational Periodic Information 92
Table 25: Conditional Periodic Information 94
Table 26: Error States 94

List of Figures

Figure 1 Block Diagram 7
Figure 2 FIPS Applet on RookySE Chip 8

1 General

1.1 Overview

This document defines the Security Policy for FIPS Applet on RookySE with firmware FIPS Applet v1.B.1.7 on RookySE '097154' cryptographic module. FIPS Applet on RookySE is a Hardware Security Module made by IDEMIA, hereafter denoted the module.

1.2 Security Levels

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

Table 1: Security Levels

1.3 Additional Information

Acronym	Definition
KCV	Key Check Value
CCMK	Crypto Card Master Key
DGI	Data Grouping Identifier
APDU	Application Protocol Data Unit
SE	Secure Element
SPI	Serial Peripheral Interface
GP	Global Platform
RAM	Random Access Memory
JCVM	Java Card Virtual Machine
DAP	Data Authentication Pattern
JCVM	Java Card Virtual Machine
JPATCH	Java Card Virtual Machine Patch

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

FIPS Applet on RookySE is a cryptographic module intended to be used as hardware security module. It is designated for creating, storing, and operating keys with some cryptographic operations capabilities, and it relies on secure element hardware with tamper-protection.

Module Type: Hardware

Module Embodiment: Single Chip

Cryptographic Boundary:

This cryptographic module is a hardware module and is a single chip. It is operated by embedded Global Platform OS with card manager capability for firmware (applet) loading, installation, or deletion. FIPS Applet is loaded at manufacturing and is part of the module.

This cryptographic module is the SLC37ESA2M0 Secure Element, defined by its physical package boundary. The TOEPP includes the host platform used during testing, consisting of a general-purpose microcontroller or microprocessor board providing power, clock, and communication interfaces (ISO7816-3 and SPI). The TOEPP also includes the associated operating system (if applicable), device drivers, and vendor-supplied

middleware that enable communication with the Secure Element. These TOEPP components are outside the module's cryptographic boundary but part of the tested configuration. Logical communication occurs via the APDU Commands, while physical communication occurs over the ISO7816-3/SPI interface. The module boundary is shown in red in the following picture:

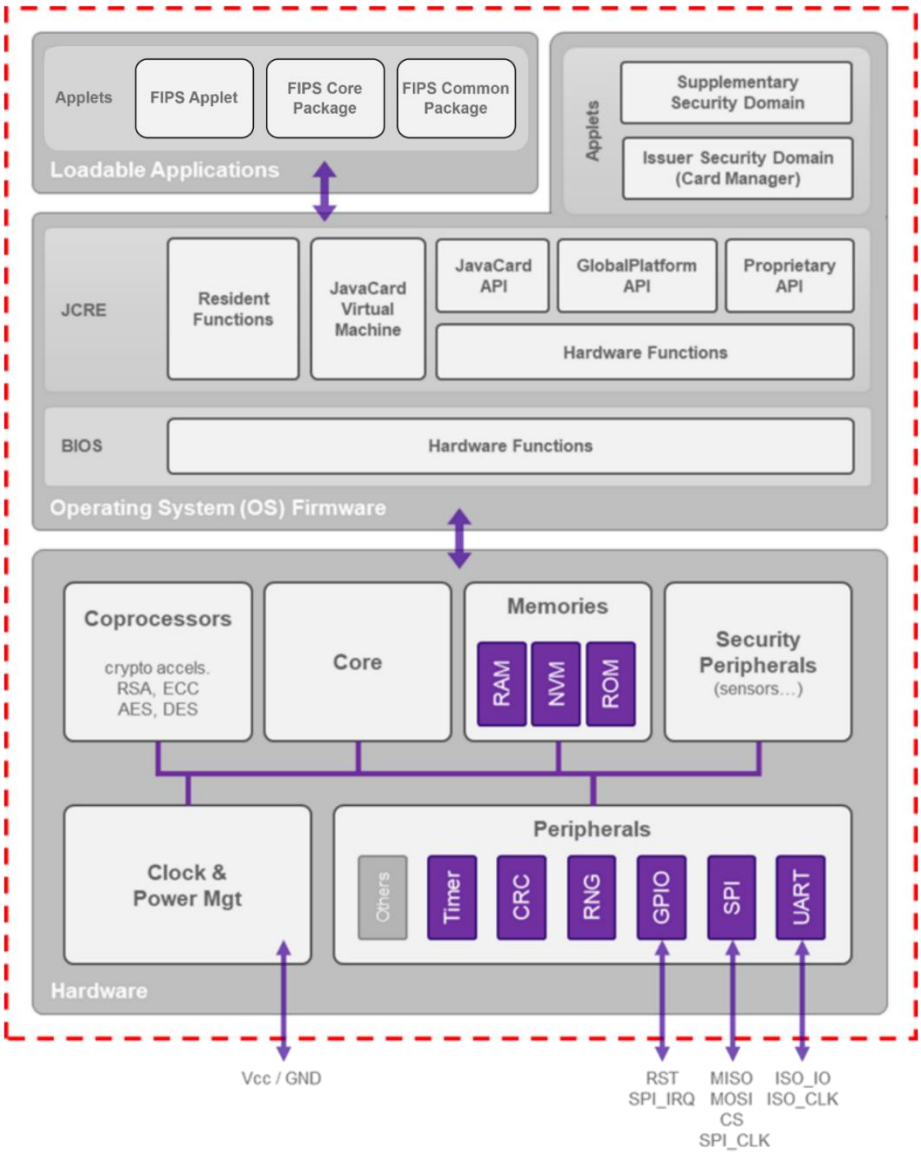


Figure 1 Block Diagram

The picture below shows the FIPS Applet on RookySE Cryptographic Module in a single chip (Module count is 1) in VQFN32 form factor with dual interface (ISO 7816 T0 Contact Protocol and SPI Protocol):

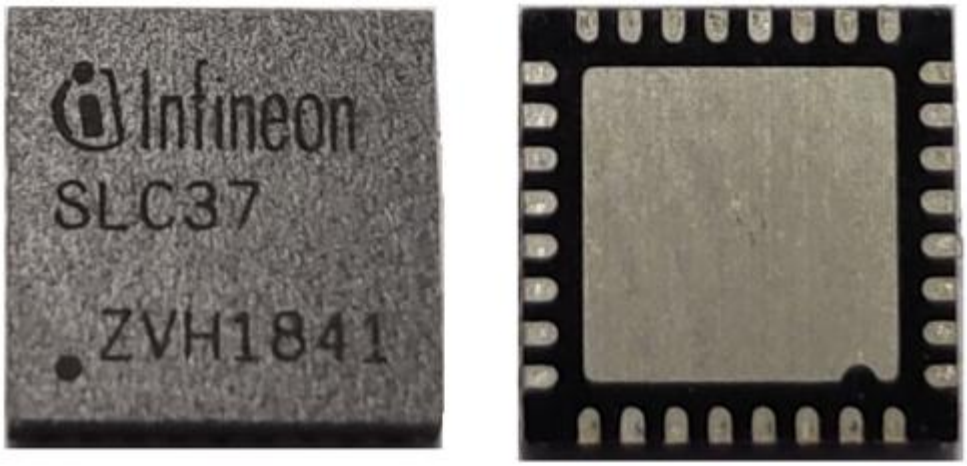


Figure 2 FIPS Applet on RookySE Chip

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
SLC37ESA2M0	Version: '29'	Global Platform OS: RookySE '097154' - Javacard Application: FIPS Applet v1.B.1.7	32-bit ARM® SecurCore® SC300™	N/A

Table 2: Tested Module Identification – Hardware

The module can be in one of the two configurations below. The configuration is set in the manufacturing stage.

FIPS Certified Product (FCP)

This product configuration is intended to meet FIPS requirements and be validated by validation authority. This Security Policy describes this configuration.

Non-FIPS Certified Product (NFCP)

This product configuration is not intended to meet FIPS requirements and is out of scope of the FIPS evaluation. In this configuration, the product supports algorithms used in the Telecommunication Industry such as MILENAGE (Rijndael), TUAK (Keccak), and ECIES for Profile A and B and other non-approved algorithms accessed by non-approved services.

The module does not implement any Vendor Affirmed Operational Environments.

2.3 Excluded Components

None

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
FIPS Certified Product (FCP)	Approved mode of operation	Approved	Global indicator via GET INFO service

Table 3: Modes List and Description

The module supports an applet instance that is running only in one mode of operation that is an approved mode. All services provided by the module when set in configuration FCP are approved services as specified in section 4.3 Approved Services. A global indicator via FIPS Applet GET INFO service (unauthenticated service) is provided to show the status mode of operation.

In FCP configuration, the module applies strictly rules of the FIPS requirements and the security policy is enforced. Access to some services is restricted and only approved services as specified in section 4.3 Approved Services are available for users. With this configuration, customers are still allowed to load additional firmware, but this is restricted to firmware (applet) that is already validated under FIPS 140-3. Any firmware (applet) that is not validated FIPS 140-3 will cancel the FIPS status of the module. A procedure for firmware loading is described in [FQR 401 9097 Ed 9] section 3.4.2.

The module does not implement any degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Note that the full cryptographic algorithm implementation capabilities were tested for the Approved cryptographic functions, but only algorithms / mode / key sizes / functionalities identified in the table below are implemented by the module.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A8099	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A8099	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-14400 Increment 8	SP 800-38B
AES-CTR	A8099	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 1-128 Supports Counter larger than maximum value - Yes Incremental Counter - Yes Conformances - RFC3686	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
		Counter Tests Performed - Yes IV Generation Mode - External	
AES-ECB	A8099	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A8099	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256 Tag Length - 128 IV Length - IV Length: 96 Payload Length - Payload Length: 0-11176 Increment 8 AAD Length - AAD Length: 0-3968 Increment 8	SP 800-38D
AES-KW	A8099	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-2048 Increment 64	SP 800-38F
AES-KWP	A8099	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-2048 Increment 8	SP 800-38F
Counter DRBG	A8099	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 808 Nonce - Nonce: 408 Personalization String Length - Personalization String Length: 128 Returned Bits - 1024	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A8099	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A8099	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A8099	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A8099	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A8099	MAC - MAC: 160 Key Length - Key Length: 8-512 Increment 8 Message Length - Message Length: 0-4096 Increment 8	FIPS 198-1
HMAC-SHA2-256	A8099	MAC - MAC: 256 Key Length - Key Length: 8-512 Increment 8 Message Length - Message Length: 0-4096 Increment 8	FIPS 198-1
HMAC-SHA2-384	A8099	MAC - MAC: 384 Key Length - Key Length: 8-512 Increment 8 Message Length - Message Length: 0-4096 Increment 8	FIPS 198-1
HMAC-SHA2-512	A8099	MAC - MAC: 512 Key Length - Key Length: 8-512 Increment 8 Message Length - Message Length: 0-4096 Increment 8	FIPS 198-1
KAS-ECC Sp800-56Ar3	A8099	Domain Parameter Generation Methods - P-521 Function - Full Validation, Key Pair Generation, Key Regeneration iutId - E3858A7CCB5348D2217E3314560FD759E3858A7CCB5348D2217E3314560FD759 Scheme - fullUnified - KAS Role - Responder KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - HMAC-SHA2-256 MAC Salting Methods - random Fixed Info Pattern - label||context||algorithmId||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Confirmation Methods - Key Confirmation Directions - Key Confirmation Roles - Key Confirmation MAC Methods - CMAC - Key Length - 256 MAC Length - 128 Key Length - 264	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A8099	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - onePassDh - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA OneStep	A8099	Auxiliary Function Methods - Auxiliary Function Name - SHA2-224 Fixed Info Pattern - t||uPartyInfo||vPartyInfo||l Fixed Info Encoding - concatenation	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
SP800-56Cr2		Derived Key Length - 528 Shared Secret Length - Shared Secret Length: 224, 256, 384, 528	
KDF SP800-108	A8099	KDF Mode - Counter MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 Supported Lengths - Supported Lengths: 64, 128, 160, 192, 224, 256, 320, 384, 512 Fixed Data Order - After Fixed Data, Before Fixed Data, In the Middle of Fixed Data Counter Length - 16, 24, 32, 8 Supports Empty IV - No Requires Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
KTS-IFC	A8099	Function - keyPairGen IUT ID - E3858A7CCB5348D2217E3314560FD759E3858A7CCB5348D2217E3314560FD759 Modulo - 2048, 3072, 4096 Key Generation Methods - rsakpg1-basic, rsakpg1-crt 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, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Key Length - 1024	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-5)	A8099	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096 p mod 8 - 0 Primality Tests - 2pow100 q mod 8 - 0 Info Generated By Server - No Private Key Format - crt Public Exponent Mode - random	FIPS 186-5
RSA SigGen (FIPS186-5)	A8099	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1	FIPS 186-5
RSA SigVer (FIPS186-4)	A8099	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096 Hash Pair -	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		Hash Algorithm - SHA-1 Public Exponent Mode - Random	
RSA SigVer (FIPS186-5)	A8099	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1 Public Exponent Mode - random	FIPS 186-5
SHA-1	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-224	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-256	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-384	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-512	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA3-224	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-256	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-384	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-512	A8099	Message Length - Message Length: 0-65528 Increment 8	FIPS 202

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

The module implements the vendor-affirmed algorithm in the table below.

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric and Asymmetric	N/A	SP800-133r2 section 4, example 1

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
CSPs obfuscation	N/A	CSPs obfuscation with a non-Approved algorithm

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

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

The module implements the security functions listed in the table below.

Name	Type	Description	Properties	Algorithms
AES	BC-UnAuth	AES encryption and decryption (ECB, CBC, CTR)		AES-CBC: (A8099) AES-ECB: (A8099) AES-CTR: (A8099)
MAC	MAC	MAC Generation/Verification		AES-CMAC: (A8099) HMAC-SHA2-512: (A8099) HMAC-SHA-1: (A8099) HMAC-SHA2-256: (A8099) HMAC-SHA2-384: (A8099)
AES GCM	BC-Auth	AES authenticated encryption and decryption (GCM)		AES-GCM: (A8099)
AES Wrap	KTS-Wrap	Key wrap with or without padding		AES-KW: (A8099) AES-KWP: (A8099)
AES Unwrap	KTS-Unwrap	Key unwrap with or without padding		AES-KW: (A8099) AES-KWP: (A8099)
DRBG	DRBG	Deterministic Random Bit Generation		Counter DRBG: (A8099)
ECDSA KeyGen	AsymKeyPair-KeyGen	ECDSA Key Generation		ECDSA KeyGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-5): (A8099)
ECDSA KeyVer	AsymKeyPair-KeyVer	ECDSA Key Verification		ECDSA KeyVer (FIPS186-5): (A8099)
ECDSA SigGen	DigSig-SigGen	ECDSA Signature Generation		ECDSA SigGen (FIPS186-5): (A8099)
ECDSA SigVer	DigSig-SigVer	ECDSA Signature Verification		ECDSA SigVer (FIPS186-5): (A8099)
KAS ECC	KAS-Full	Key Agreement	IG:IG D.F Scenario 2, path (2), end-to-end Key confirmation:Yes Key derivation :KDA (tested as part of KAS certificate) Caveat:Key establishment methodology provides 256 bits of security strength	KAS-ECC Sp800-56Ar3: (A8099)
KDF	KBKDF	Symmetric Keys Derived from a Pre-existing Key using KDF		KDF SP800-108: (A8099)
KAS SSC	KAS-SSC	Shared secret computation	IG:IG D.F Scenario 2, path (1)	KAS-ECC-SSC Sp800-56Ar3: (A8099)
KDA	KAS-56CKDF	Derive symmetric encryption and MAC keys from a shared secret.		KDA OneStep SP800-56Cr2: (A8099)
GP KTS Wrap	KTS-Wrap	Key wrapping for Secure Channel GP	IG D.G:approved method from IG D.G Caveat:Key establishment	AES-CBC: (A8099) Key length: 256 bits AES-CMAC: (A8099)

Name	Type	Description	Properties	Algorithms
			methodology provides 256 bits of security strength	Key length: 256 bits
GP KTS Unwrap	KTS-Unwrap	Key unwrapping for Secure Channel GP	IG D.G:Approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-CBC: (A8099) Key length: 256 bits AES-CMAC: (A8099) Key length: 256 bits
SC KTS Wrap	KTS-Wrap	Key wrapping for Secure Channel	IG D.G:Approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-CBC: (A8099) Key length: 256 bits AES-CMAC: (A8099) Key length: 256 bits
SC KTS Unwrap	KTS-Unwrap	Key unwrapping for Secure Channel	IG D.G:Approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-CBC: (A8099) Key length: 256 bits AES-CMAC: (A8099) Key length: 256 bits
Token KTS Wrap	KTS-Wrap	Key wrapping for Token transport	IG D.G:Approved method from IG D.G Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A8099) AES-CMAC: (A8099)
Token KTS Unwrap	KTS-Unwrap	Key unwrapping for Token transport	IG D.G:Approved method from IG	AES-CBC: (A8099)

Name	Type	Description	Properties	Algorithms
			D.G Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CMAC: (A8099)
KTS Encap	KTS-Encap	RSA-OAEP encryption for Token transport	IG D.G: Caveat: Key establishment methodology provides between 128 and 256 bits of security strength Key confirmation:No	KTS-IFC: (A8099)
KTS Decap	KTS-Decap	RSA-OAEP decryption for Token transport	IG D.G:Approved method from IG D.G Caveat: Key establishment methodology provides between 112 and 128 bits of security strength Key confirmation:No	KTS-IFC: (A8099)
RSA KeyGen	AsymKeyPair-KeyGen	RSA Key Generation		RSA KeyGen (FIPS186-5): (A8099)
RSA SigGen	DigSig-SigGen	RSA Signature Generation using PCKS1 v1.5 and PSS Scheme		RSA SigGen (FIPS186-5): (A8099)
RSA SigVer	DigSig-SigVer	RSA Signature Verification using PCKS1 v1.5 and PSS Scheme		RSA SigVer (FIPS186-4): (A8099) RSA SigVer (FIPS186-5): (A8099)
SHA	SHA	Message digest		SHA-1: (A8099) SHA3-512: (A8099)

Name	Type	Description	Properties	Algorithms
				SHA2-384: (A8099) SHA3-256: (A8099) SHA3-224: (A8099) SHA2-224: (A8099) SHA2-512: (A8099) SHA2-256: (A8099) SHA3-384: (A8099)
CKG	CKG	Symmetric key generation		CKG: () Key Type: Symmetric
Entropy source	ENT-ESV	Request Entropy		

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM: The implementation only supports 96-bit random IV, provided by the approved DRBG defined in section 2.6 Security Function Implementations, following IG C.H scenario 2.

SHA-1: Its use is disallowed for digital signature generation and deprecated through December 31, 2030 for non-digital signature applications.

RSA: Signature verification using 1024-bit modulus is only permitted for legacy use.

2.8 RBG and Entropy

The RBG source within this module is from entropy source specified in the Table Approved Algorithm.

Cert Number	Vendor Name
E240	Infineon Technologies AG

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Infineon 32-bit Security Controller Entropy Source 2	Physical	SLC 37 32-bit SecurityController V07	8	2.844587	N/A

Table 9: Entropy Sources

All configuration of the entropy source is automatically set by the module at startup.

The entropy source output is used to seed DRBG. The uses of DRBG output in the module are described below:

1. Generate random number for initial Nonce for key protection by CCMK during FIPS Applet Installation
2. Generate key for CCMK in “CCMK Export 1” service
3. Generate random numbers for Split procedure in “CCMK export” services and “Generate Key (for Split Knowledge)” service
4. Generate key for Customer key in “Generate Key (for Split Knowledge)” service
5. Generate key for Symmetric Key in “Generate Key” Service
6. Generate key for Asymmetric Key (RSA and EC Key pair) in “Generate Key Pair” Service
7. Ephemeral EC Key Generation during Key Agreement in the Mutual Authentication service
8. Utilized in Cipher process of RSA PKCS and PSS
9. Utilized in Signature Generation process using ECDSA
10. Utilized to generate card challenge in GP Initialize Update command
11. Utilized in GP Put Key Command
12. Utilized in the creation of OS-MKEK
13. Utilized in Initialization Vector (IV) Generation for AES-GCM Encryption

The approved services that use DRBG are listed in the section 4.3 Approved Services.

2.9 Key Generation

Asymmetric RSA and ECDSA keys are generated in accordance with FIPS 186-5. Symmetric keys are generated in accordance with SP800-133rev2 section 4 example 1.

2.10 Key Establishment

The cryptographic module supports the following key agreement mechanisms:

1. Elliptic Curve Diffie-Hellman (ECDH)
   - Algorithm: ECDH KAS ECC C(2e, 2s)
   - Elliptic Curve Parameters: NIST P-521
   - Key derivation: One Step KDF based on HMAC-SHA2-256
   - Key Confirmation: Yes (based on CMAC)
   - Security Strength: 256 bits
   - Approval Status: Approved per SP 800-56A Rev. 3
2. Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)
   - Algorithm: ECDHE KAS ECC SSC C(1e, 1s)
   - Elliptic Curve Parameters: NIST P224, P-256, P-384, P-521
   - Key derivation: One Step KDF based on SHA2 or SHA3 (separately tested)
   - Key Confirmation: No
   - Security Strength: Between 112 to 256 bits (depending on curve)
   - Approval Status: Approved per SP 800-56A Rev. 3

The cryptographic module supports the following key transport mechanisms:

1. RSA OAEP
   - Algorithm: RSAES-OAEP
   - Key Sizes: 2048, 3072, 4096 bits
   - Padding Scheme: OAEP with SHA
   - Security Strength: Between 112 and 128 bits
   - Approval Status: Approved per SP 800-56B Rev. 2
2. AES Key Wrapping
   - Algorithm: AES Key Wrap (KW and KWP)
   - Key Sizes: 128, 192, 256 bits
   - Security Strength: Up to 256 bits
   - Approval Status: Approved per SP 800-38F
3. KTS
   - Description: KTS (key wrapping and unwrapping) per IG D.G.
   - Algorithm: AES-CBC, AES-CMAC
   - Key Sizes: 128, 192, 256 bits
   - Security Strength: Up to 256 bits
   - Approval Status: Approved per SP 800-38F

2.11 Industry Protocols

None

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The module provides a dual interface for communications that is available to all users: ISO7816 T0 Contact Protocol and Serial Peripheral Interface (SPI) Protocol. These two interfaces cannot be used simultaneously. Data output is inhibited during error states in any interface. The module acts as a slave device and does not have control output.

Physical Port	Logical Interface(s)	Data That Passes
Vcc, GND	Power	Supply voltage
RST	Control Input	Chip Reset
CLK	Control Input	ISO 7816: Clock
I/O	Data Input Data Output Control Input Status Output	ISO 7816: Input / Output of Data, Status Word, Procedure Byte
SPI_CLK	Control Input	SPI: Clock
SPI_MISO	Data Input Control Input	SPI: Master IN, Slave OUT

Physical Port	Logical Interface(s)	Data That Passes
SPI_MOSI	Data Output Status Output	SPI: Master OUT, Slave IN
SPI_CS	Control Input	SPI: Chip Select
SPI_IRQ	Status Output	SPI: A status signal to inform command completion

Table 10: Ports and Interfaces

3.2 Additional Information

In the ISO7816 T0 protocol, data output is inhibited during key generation, self-tests, and zeroisation except for procedure byte NULL '60' transmission to keep the communication between module and the interface device still alive.

In the SPI protocol, data output is inhibited during key generation, self-tests, and zeroisation with exception for Check Alive SPI command. This command is used to check if the module is still alive or not. This command can temporarily interrupt the current execution of key generation, self-tests, or zeroisation.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module supports the authentication methods listed in the table below.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
GP SCP '03' Authentication	The Secure Channel Protocol authentication method is provided by the Secure Channel service. The SD-KENC and SD-KMAC keys are used to derive the SD-SENC and SD-SMAC keys, respectively. The off-card entity participating in the mutual authentication sends a 64-bit challenge to the Cryptographic	MAC	Based on the the size of the MAC, the probability that a random attempt will succeed using this authentication method is 1/(2^128)	The module enforces a "slowdown mechanism" that increases the response time between two authentications attempts following a failed authentication, such that no more than 9 attempts are possible in a one-minute period. The probability that a random attempt will succeed over a one-

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	Module. The Cryptographic Module generates its own challenge and computes a 64-bit cryptogram with SD-SMAC key and both challenges. The Cryptographic Module cryptogram and challenge are sent to the off-card entity which checks the Cryptographic Module's cryptogram and creates its own 64-bit cryptogram with both challenges. A 64-bit message authentication code (MAC) is also computed on the command containing the off-card entity cryptogram with AES-CMAC and SD-SMAC key, the MAC is concatenated to the command, and the command is sent to Cryptographic Module. The Cryptographic Module checks the message authentication code and compares the received cryptogram to the calculated cryptogram. If all of this succeeds, the two participants are mutually			minute interval is 9/(2^128)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	authenticated (the external entity is authenticated to the Module in the AA role). GP Secure Channel Protocol establishment provides mutual authentication service as well as establishment of a secure channel to protect confidentiality and integrity of the transmitted data.			
FIPS Applet Password-based Authentication	The operator must enter its username and its password for authentication process. The username is an alphanumeric string. The password is a binary string of a minimum of eight (8) characters. Key Agreement technique ((Cofactor) Full Unified Model, C(2e, 2s, ECC CDH) with bilateral key confirmation is used for mutual authentication and to derive session keys (H-SKAuthEnc, H-SKAuthMac, H-SKAuthKC). H-SKAuthKC is used to calculate MAC of user credential (user token) for	KAS ECC	Based on the fact that the 128-bit cryptogram is computed with a 256-bit key, the probability that a random attempt will succeed using this authentication method is 1/(2^128)	The module enforces a “slowdown mechanism” that increases the response time between two authentications attempts following a failed authentication, such that no more than 9 attempts are possible in a one-minute period. The probability that a random attempt will succeed over a one-minute interval is 9/(2^128)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	authentication. If this mutual authentication process is successful, the user can be verified via Key Confirmation using USER AUTH service employing user token. Other session keys, H-SKAuthEnc and H-SKAuthMac are used for protecting the message in Secure Channel. This scheme protects from eavesdropping and provides perfect forward secrecy.			
FIPS Applet Smartcard-based Authentication	The operator must use smartcard that owns unique user EC-key pair used for Key Agreement ((Cofactor) Full Unified Model, C (2e, 2s, ECC CDH) with bilateral key confirmation). The static user key is different per user and stored inside the smartcard which is considered as secure enclave. The user must enter his pin for verification by the smartcard to use his static key and to perform key agreement. The rest of user authentication process is similar	KAS ECC	Based on the fact that the 128-bit cryptogram is computed with a 256-bit key, the probability that a random attempt will succeed using this authentication method is 1/(2^128)	The module enforces a “slowdown mechanism” that increases the response time between two authentications attempts following a failed authentication, such that no more than 9 attempts are possible in a one-minute period. The probability that a random attempt will succeed over a one-minute interval is 9/(2^128)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	with the password-based method, except there is no password involved in the user token.			

Table 11: Authentication Methods

In the FIPS applet implementation, the module does support authentication level up to two levels where the last authenticated user will override the access control of the first authenticated user. The active access control in the second level is based on second authenticated user's profile. When this user logs off, the active access control is set back to the first authenticated user's profile. This authentication level is only applicable for authentication number 2 and number 3 above that is designed for specific procedure such as Key Ceremony, System Reset, or System Unblock services.

Global Platform Secure Channel Protocol '03' (GP SCP '03') Authentication

The Secure Channel Protocol authentication method is provided by the Secure Channel service. The SD-KENC and SD-KMAC keys are used to derive the SD-SENC and SD-SMAC keys, respectively. The off-card entity participating in the mutual authentication sends a 64-bit challenge to the Cryptographic Module. The Cryptographic Module generates its own challenge and computes a 64-bit cryptogram with SD-SMAC key and both challenges. The Cryptographic Module cryptogram and challenge are sent to the off-card entity which checks the Cryptographic Module's cryptogram and creates its own 64-bit cryptogram with both challenges. A 64-bit message authentication code (MAC) is also computed on the command containing the off-card entity cryptogram with AES-CMAC and SD-SMAC key, the MAC is concatenated to the command, and the command is sent to Cryptographic Module. The Cryptographic Module checks the message authentication code and compares the received cryptogram to the calculated cryptogram. If all of this succeeds, the two participants are mutually authenticated (the external entity is authenticated to the Module in the AA role).

GP Secure Channel Protocol establishment provides mutual authentication service as well as establishment of a secure channel to protect confidentiality and integrity of the transmitted data.

FIPS Applet Authentication Methods

The FIPS Applet uses identity-based operator authentication to enforce the separation of roles and allow corresponding services within each role.

- **FIPS Applet Password-based Authentication:**
  The operator must enter its username and its password for authentication process. The username is an alphanumeric string. The password is a binary string of a minimum of eight (8) characters. Key Agreement technique ((Cofactor) Full Unified Model, C(2e, 2s, ECC CDH) with bilateral key confirmation is used for mutual authentication and to derive session keys (H-SKAuthEnc, H-SKAuthMac, H-SKAuthKC). H-SKAuthKC is used to calculate MAC of user credential (user token) for authentication. If this mutual authentication process is successful, the user can be verified via Key Confirmation using USER AUTH service employing user token. Other session keys, H-SKAuthEnc and H-SKAuthMac are used for protecting the message in Secure Channel. This scheme protects from eavesdropping and provides perfect forward secrecy.
- **FIPS Applet Smartcard-based Authentication:**

The operator must use smartcard that owns unique user EC-key pair used for Key Agreement ((Cofactor) Full Unified Model, C (2e, 2s, ECC CDH) with bilateral key confirmation). The static user key is different per user and stored inside the smartcard which is considered as secure enclave. The user must enter his pin for verification by the smartcard to use his static key and to perform key agreement. The rest of user authentication process is similar with the password-based method, except there is no password involved in the user token.

Upon correct authentication, the role is selected based on current logged user's profile. During authentication session keys are negotiated which are used to secure subsequent services requests from operator. Since the session keys (and session ID) are stored in volatile memory all information about the authentication and session is lost if the module is powered down.

4.2 Roles

The module supports the roles listed in the table below.

Name	Type	Operator Type	Authentication Methods
Application Administrator – AA	Identity	GP Administrator	GP SCP '03' Authentication
Super User – SU	Identity	Crypto Officer	FIPS Applet Smartcard-based Authentication FIPS Applet Password-based Authentication
Administrator – CO	Identity	Crypto Officer	FIPS Applet Smartcard-based Authentication FIPS Applet Password-based Authentication
Auditor – AU	Identity	Crypto Officer	FIPS Applet Smartcard-based Authentication FIPS Applet Password-based Authentication
Key Custodians – KC	Identity	Crypto Officer	FIPS Applet Smartcard-based Authentication FIPS Applet Password-based Authentication
Standard User – UR	Identity	User	FIPS Applet Smartcard-based Authentication FIPS Applet Password-based Authentication

Table 12: Roles

Hereafter the description of each role:

Role Name	ID	Role Description
Application Administrator	**AA**	***GP Administrator*** *- This role is responsible for upgrading and loading the main firmware version of the system and additional customer applet*

		(delete/load/install Applet). This role is authenticated using Global Platform Secure Channel Protocol '03'
Super User	**SU**	***Crypto Officer Role 1*** *- Performing module Initialization for ADMIN Creation process, Unblock Administrator, System Reset using DUAL Control with Administrator* *This role is eligible to participate in Quorum Authentication. See section 4.3.*
Administrator	**CO**	***Crypto Officer Role 2*** *- Performing module initialization with the help of Super User for its creation, global configuration, user management, and profile management for user and key* *This role is eligible to participate in Quorum Authentication. See section 4.3.*
Auditor	**AU**	***Crypto Officer Role 3*** *- Performing Audit Management* *This role is eligible to participate in Quorum Authentication. See section 4.3.*
Key Custodians	**KC**	*Performing Splitting Knowledge Procedure in the Key Ceremony and hold secrets of Crypto Card Master Key (CCMK)* *This role is split into 3 different roles:* **KC1** *– manage secret 1 of CCMK in the Splitting Knowledge Procedure* **KC2** *– manage secret 2 of CCMK in the Splitting Knowledge Procedure* **KC3** *– manage secret 3 of CCMK in the Splitting Knowledge Procedure*
Standard User	**UR**	***User Role*** *– Performing general security services including Key Management and Cryptographic services* *The standard user's role is configurable based on access control list in a profile that is set by ADMIN for a user. E.g. a user can be standard user A that has role to do Crypto functions but no role for key management.* *This role is eligible to participate in Quorum Authentication. See section 4.3.*
Unauthenticated User	**UU**	A role that does not require an authentication of an operator for the role to perform some services where CSPs and PSPs are not modified, disclosed [for CSPs only], or substituted

The module does not support a maintenance role.

Additionally, any user is allowed to perform non-sensitive services such as requesting status information, without prior authentication.

4.3 Approved Services

The module only provides approved services listed in this section. The module does not provide bypass or self-initiated output capabilities. Each service of the module returns completion status to indicate successful execution or specific error code.

Some services do not require an operator to assume an authorized role. In this case the associated role is marked with the role UU (Unauthenticated User) defined in section 4.3 Approved Services.

The module implements the approved services detailed in the table below.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
GP Secure Channel	Establish a Global Platform secure communications channel	Completion Status	Key Identifier, Host Challenge, Security Level Host Cryptogram, MAC	Card Challenge, Card Cryptogram, diversification data, and key information, Status Word of the command	KDF GP KTS Wrap GP KTS Unwrap	Application Administrator – AA - OS-DRBG-STATE: G,E - SD-KENC: E - SD-KMAC: E - SD-SENC: G,E - SD-SMAC: G,E - SD-RMAC: G,E
GP Manage Content	Load and install application packages and its associated keys and data, also support Jpatch.	Completion Status	Loaded package and stored data and key encrypted through GP SCP ‘03’ Security Level 3	Status Word of the command	KDF GP KTS Wrap GP KTS Unwrap	Application Administrator – AA - SD-KENC: W - SD-KMAC: W - SD-KDEK: W - SD-SENC: E - SD-SMAC: E - SD-RMAC: E - DAP-AES : W,E - JPATCH-AES: E - JPATCH-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						AES-SENC: G,E - JPATCH-AES-SMAC: G,E
GP Get Status (Show status)	Show status service as specified in the [ISO/IEC 19790] section 7.4.3	Completion Status	Application Type	List of application and its status, Status Word of the command	None	Application Administrator – AA
GP Get Data (Show version)	Show module's versioning information service as specified in the ISO19790: 2012 section 7.4.3	Completion Status	Data Tag	Corresponding data based on input tag	None	Application Administrator – AA
GP Life Cycle	Set GP life cycle	Completion Status	Status type and State Control	Status Word of the command	None	Application Administrator – AA - OS-MKEK: Z
GP Select	There is no intervention to the security process nor access to defined SSP	Completion Status	Application (applet) AID	Status Word of the command	None	Unauthenticated
SE Reset	Reset Warm/Cold and zeroize	Completion Status	Signal Reset in port RST	Answer-to-RESET (ATR)	None	Unauthenticated - H-eAUTH_sk: Z - H-Zs: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-Ze: Z - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: Z - H-SKSyncEnc: Z - H-SKSyncMac: Z - H-eAUTH_pk: Z - H-eHOST_pk: Z - H-HostID: Z - E-eEC_KAS_pk: Z - H-eEC_KAS_sk: Z - H-eEC_KAS_pk: Z - H-Zd: Z
Initialize Update	Performs initiation of a GP SCP '03' Secure Channel Session	Completion Status	Key Identifier and host challenge	Card Challenge, Card Cryptogram, diversification data, and key information, Status Word of the command	KDF GP KTS Wrap GP KTS Unwrap	Application Administrator – AA - SD-SENC: G,E - SD-SMAC: G,E - SD-RMAC: G
External	Authenticate the host and to	Completion Status	Security Level 3, Host Cryptogram and MAC	Status Word of the	GP KTS Wrap	Application Administrator – AA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Authenticate	determine the level of security required for all subsequent commands			command	GP KTS Unwrap	- SD-SENC: E - SD-SMAC: E - SD-RMAC: E
Store Data	Transfer data to an Application in the GP secure channel	Completion Status	Sensitive Data and Key encrypted through GP SCP '03' Security Level 3	Status Word of the command	AES GP KTS Wrap GP KTS Unwrap	Application Administrator – AA - H-sAUTH_sk: W - H-KSync: W - H-sAUTH_pk: W - H-sHOST_pk: W - H-sUSER_pk: W
Get Info (Show version)	Show module's versioning information and Show status service as specified in the [ISO/IEC 19790] section 7.4.3	Completion Status	Information type	System information as requested such as: Applet version, applet build type, single/multi SE configuration, life cycle status, storage size configuration, TC and sync status, current	None	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
				active user, and/or last executed frame id Return Status		
Manage Session	FIPS Applet open session and close session	Completion Status	Session type	Return Status	None	Unauthenticated - H-KT_ECDSA_PAIR: Z - H-KT_ECDSA_sk: Z - H-KT_AES: Z - H-KT_AES_CMAC: Z - H-KT_AES_WRAP: Z - H-KT_AES_GCM: Z - H-KT_3DES: Z - H-KT_RSA_CRT_PAIR: Z - H-KT_RSA_SFM_PAIR: Z - H-KT_RSA_CRT_sk: Z - H-KT_RSA_SFM_sk: Z - H-KT_HMAC: Z - H-KT_ECDSA_p

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						k: Z - H-sEC_KAS_pk: Z - H-sEC_KAS_sk: Z - H-KT_RSA_pk: Z - H-KT_RSA_OAEP_PAIR: Z - H-KT_RSA_OAEP_sk: Z - H-KT_RSA_OAEP_pk: Z - E-sEC_KAS_pk: Z - H-Zd: Z
Mutual Authenticate	Public key exchange leads to key agreement between HOST's operator and the module	Completion Status	username, user/host public ephemeral key, selected authentication method, and host ID	System's public ephemeral key and receipt for key confirmation Return Status	KAS ECC	Unauthenticated - H-sAUTH_sk: E - H-eAUTH_sk: G,E,Z - H-Zs: G,E,Z - H-Ze: G,E,Z - H-SKAuthEnc: G - H-SKAuthMac: G - H-SKAuthKC: G,E - H-sAUTH_pk: E - H-sHOST_pk: E - H-sUSER_pk: E - H-eAUTH_pk: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-eHOST_pk: G,E,Z - H-HsmID: E - H-HostID: W,E
User Authenticate	Login on current open session	Completion Status	user token generated from MAC of user's credential using generated session key from Key Agreement process in Mutual Authenticate	Authentication result Return Status	MAC	Unauthenticated - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: E,Z - H-User_pwd: E
User Logout	Logout the user	Completion Status	Username information	Return Status	None	Super User – SU - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: Z Administrator – CO - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: Z Key Custodians – KC - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: Z Auditor – AU - H-SKAuthEnc: Z - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						SKAuthMac: Z - H-SKAuthKC: Z Standard User – UR - H-SKAuthEnc: Z - H-SKAuthMac: Z - H-SKAuthKC: Z
Secure Channel	Secure Messaging for sensitive data	Completion Status	Encrypted data and its signature	Encrypted response data Return Status	AES MAC	Super User – SU - H-SKAuthEnc: E,Z - H-SKAuthMac: E,Z - H-SKAuthKC: E,Z Administrator – CO - H-SKAuthEnc: E,Z - H-SKAuthMac: E,Z - H-SKAuthKC: E,Z Key Custodians – KC - H-SKAuthEnc: E,Z - H-SKAuthMac: E,Z - H-SKAuthKC: E,Z Auditor – AU - H-SKAuthEnc:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E,Z - H-SKAuthMac: E,Z - H-SKAuthKC: E,Z Standard User – UR - H-SKAuthEnc: E,Z - H-SKAuthMac: E,Z - H-SKAuthKC: E,Z
Check Log	Return N last logs based user input	Completion Status	Number of logs to be returned	Activity logs and Return Status	None	Auditor – AU
Read Log	Return and remove N last logs based user input	Completion Status	Number of logs to be read and deleted	Activity logs and Return Status	None	Auditor – AU
Delete Log	Delete N last logs based user input	Completion Status	Number of logs to be deleted	Return Status	None	Auditor – AU
Export Log (mode 01)	Export all logs in the log database(all logs deleted)	Completion Status	Mode to export the logs (mode: deleted)	All activity logs and Return Status	None	Auditor – AU
Export Log (mode 02)	Export all logs in the log database(all logs kept)	Completion Status	Mode to export the logs (mode: kept)	All activity logs and Return Status	None	Auditor – AU Administrator – CO
Profile Create	Create new profile	Completion Status	Profile name, access control list, max number of users	Return Status	None	Administrator – CO

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			(N)(optional), quorum number (K)(optional), and quorum auth timeout (optional)			
Profile Delete	Delete non-default profile based on profile name input	Completion Status	Profile name and deletion option	Return Status	None	Administrator – CO
Profile View	Get the list of all profiles with their respective ACL within FIPS Applet	Completion Status	Only input frame with no additional input data	information of all profiles Return Status	None	Administrator – CO
User Create	Create new user	Completion Status	User information and credential data	Return Status	ECDSA SigVer	Administrator – CO - H-User_pwd: W - H-sUSER_pk: W
User Create ADMIN	Admin creation. This is a step during FIPS Applet Initialization procedure	Completion Status	Admin's information and credential data	Return Status	ECDSA SigVer	Super User – SU - H-User_pwd: W - H-sUSER_pk: W
User Delete	Delete user based on username input	Completion Status	Username information	Return Status	ECDSA SigVer	Administrator – CO - H-User_pwd: Z - H-sUSER_pk: Z
User Change	Change user's password	Completion Status	Username and new user's password or new user's public key	Return Status	ECDSA SigVer	Super User – SU - H-User_pwd:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Credential	or user's public key					W - H-sUSER_pk: W Administrator – CO - H-User_pwd: W - H-sUSER_pk: W
User View (full)	Return the list of all credentials within FIPS Applet	Completion Status	User Type (All Users)	Information of all users Return Status	None	Administrator – CO
User View (Admin only)	Return the list of admin credentials within FIPS Applet	Completion Status	User Type (Admin only)	information about admin user Return Status	None	Super User – SU Administrator – CO
CCMK Import 1	Import secret based on user's (KeyCustodian1) input	Completion Status	Secret data and its key check value to import CCMK using split knowledge procedure	Return Status	None	Administrator – CO Key Custodians – KC
CCMK Import 2	Import secret based on user's (KeyCustodian2) input	Completion Status	Secret data and its key check value to import CCMK using split knowledge procedure	Return Status	None	Administrator – CO Key Custodians – KC
CCMK Import 3	Import secret based on user's (KeyCustodian3) input. CCMK will be set in	Completion Status	Secret data and its key check value to import CCMK using split knowledge procedure	Key check value of CCMK Return Status	MAC KDF	Administrator – CO - H-CCMK: W,Z - H-CCMKEnc: G,Z - H-CCMKMac: G,Z Key

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	this sequence					Custodians – KC - H-CCMK: W,Z - H-CCMKEnc: G,Z - H-CCMKMac: G,Z
CCMK Export 1	Export secret of KeyCustodian1 Secret is calculated by FIPS applet in this case	Completion Status	Only input frame with no additional input data	Secret data and its key check value of CCMK using split knowledge procedure Return Status	None	Administrator – CO - H-CCMK: G Key Custodians – KC - H-CCMK: G
CCMK Export 2	Export secret of KeyCustodian2. Secret is calculated by FIPS applet in this case	Completion Status	Only input frame with no additional input data	Secret data and its key check value of CCMK using split knowledge procedure Return Status	None	Administrator – CO - H-CCMK: E Key Custodians – KC - H-CCMK: E
CCMK Export 3	Export secret of KeyCustodian3. Secret is calculated by FIPS	Completion Status	Only input frame with no additional input data	Secret data and its key check value of CCMK using	MAC KDF	Administrator – CO - H-CCMK: E - H-CCMKEnc: G - H-CCMKMac: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	applet in this case. CCMK will be set in this sequence			split knowledge procedure Return Status		Key Custodians – KC - H-CCMK: E - H-CCMKEnc: G - H-CCMKMac: G
System Set Config	Set FIPS applet configuration	Completion Status	00 to keep FIPS Certified Product 01 to change to Non FIPS Certified Product	Return Status	None	Administrator – CO
Import Key	Create new keys with parameters totally based on user's input.	Completion Status	Imported key in key token form protected by protection key whose label specified in key token's properties. Unwrap key in key token form protected by protection key whose label specified in key token's properties. For CKM_ECDH_AES_KEY_WRAP, Ephemeral Public Transport Key shall be concatenated with wrapped key.	Imported key in key token form protected by CCMK of the module as protection key Return Status	AES MAC AES Unwrap KAS SSC SC KTS Unwrap Token KTS Unwrap KTS Decap	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: W,E - H-KT_AES_CMAC: W,E - H-KT_AES_WRAP: W,E - H-KT_AES_GCM: W - H-KT_3DES: W - H-KT_HMAC: W - H-KT_RSA_CRT_PAIR: W - H-KT_RSA_SFM_PAIR: W - H-KT_RSA_pk: W - H-KT_RSA_CRT

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						_sk: W - H-KT_RSA_SFM_sk: W - H-KT_ECDSA_PAIR: W - H-KT_ECDSA_pk: W - H-KT_ECDSA_sk: W - H-sEC_KAS_sk: W,E - E-eEC_KAS_pk: W,E,Z - H-KT_SecretData: W - H-KT_RSA_OAEP_PAIR: W,E - H-KT_RSA_OAEP_sk: W,E - H-KT_RSA_OAEP_pk: W - H-Zd: G,E,Z - H-eWrap: G,E,Z Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: W,E - H-KT_AES_CMAC: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_AES_WRAP: W,E - H-KT_AES_GCM: W - H-KT_3DES: W - H-KT_HMAC: W - H-KT_RSA_CRT_PAIR: W - H-KT_RSA_SFM_PAIR: W - H-KT_RSA_pk: W - H-KT_RSA_CRT_sk: W - H-KT_RSA_SFM_sk: W - H-KT_ECDSA_PAIR: W - H-KT_ECDSA_pk: W - H-KT_ECDSA_sk: W - H-sEC_KAS_sk: W,E - E-eEC_KAS_pk: W,E,Z - H-KT_SecretData: W - H-KT_RSA_OAE

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						P_PAIR: W,E - H-KT_RSA_OAEP_sk: W,E - H-KT_RSA_OAEP_pk: W - H-Zd: G,E,Z - H-eWrap: G,E,Z
Export Key	Export existing key into key file with CCMK or other existing key protection	Completion Status	Exported key label and protection key label where those key tokens shall be loaded to the module prior to this service via Load Key service. The to-be wrapped key label and the wrapping key label where those key tokens shall be loaded to the module prior to this service via Load Key service	For Export Key: Exported key in key token form with protection key as configured in input frame For Wrap Key: Wrapped key data Return Status	AES MAC AES Wrap KAS SSC SC KTS Wrap Token KTS Wrap KTS Encap	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: R,E - H-KT_AES_CMAC: R,E - H-KT_AES_WRAP: R,E - H-KT_AES_GCM: R - H-KT_3DES: R - H-KT_HMAC: R - H-KT_RSA_CRT_PAIR: R - H-KT_RSA_SFM_PAIR: R - H-KT_RSA_pk: R - H-KT_RSA_CRT_sk: R - H-KT_RSA_SFM_sk: R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_ECDSA_PAIR: R - H-KT_ECDSA_pk: R - H-KT_ECDSA_sk: R - H-eEC_KAS_pk: G,R,Z - H-eEC_KAS_sk: G,E,Z - E-sEC_KAS_pk: E - H-KT_SecretData: R - H-KT_RSA_OAEP_PAIR: R,E - H-KT_RSA_OAEP_pk: R,E - H-KT_RSA_OAEP_sk: R - H-Zd: G,E,Z - H-eWrap: G,E,Z - H-sEC_KAS_pk: R Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: R,E - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KT_AES_CMAC: R,E - H-KT_AES_WRAP: R,E - H-KT_AES_GCM: R - H-KT_3DES: R - H-KT_HMAC: R - H-KT_RSA_CRT_PAIR: R - H-KT_RSA_SFM_PAIR: R - H-KT_RSA_pk: R - H-KT_RSA_CRT_sk: R - H-KT_RSA_SFM_sk: R - H-KT_ECDSA_PAIR: R - H-KT_ECDSA_pk: R - H-KT_ECDSA_sk: R - H-eEC_KAS_pk: G,R,Z - H-eEC_KAS_sk: G,E,Z - E-sEC_KAS_pk: E - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KT_SecretData: R - H-KT_RSA_OAEP_PAIR: R,E - H-KT_RSA_OAEP_pk: R,E - H-KT_RSA_OAEP_sk: R - H-Zd: G,E,Z - H-eWrap: G,E,Z - H-sEC_KAS_pk: R
Load Key	Load PERMANENT key from LKD or VOLATILE FOR STORAGE key from LKD_DP into FIPS applet	Completion Status	Loaded key in key token form protected by protection key whose label specified in key token's properties	Return Status	AES Unwrap KAS SSC Token KTS Unwrap KTS Decap	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: W - H-KT_AES_CMAC: W - H-KT_AES_WRAP: W - H-KT_AES_GCM: W - H-KT_3DES: W - H-KT_HMAC: W - H-KT_RSA_CRT_PAIR: W - H-KT_RSA_SFM_PAIR: W - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KT_RSA_pk: W - H-KT_RSA_CRT_sk: W - H-KT_RSA_SFM_sk: W - H-KT_ECDSA_PAIR: W - H-KT_ECDSA_pk: W - H-KT_ECDSA_sk: W - H-KT_SecretData: W - E-sEC_KAS_pk: W - H-sEC_KAS_sk: W Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: W - H-KT_AES_CMAC: W - H-KT_AES_WRAP: W - H-KT_AES_GCM: W - H-KT_3DES: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_HMAC: W - H-KT_RSA_CRT_PAIR: W - H-KT_RSA_SFM_PAIR: W - H-KT_RSA_pk: W - H-KT_RSA_CRT_sk: W - H-KT_RSA_SFM_sk: W - H-KT_ECDSA_PAIR: W - H-KT_ECDSA_pk: W - H-KT_ECDSA_sk: W - H-KT_SecretData: W - E-sEC_KAS_pk: W - H-sEC_KAS_sk: W
Diversify Key	Diversify an existing key with user's choice of method	Completion Status	Key token properties and diversification method or derivation method depends on Operation mode chosen	Diversified key or Derived Key in key token form protected by CCMK of the module	KDF KDA	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G,E,R - H-KT_AES_CMA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
				as its protection key Return Status		C: G,E,R - H-KT_AES_WRAP: G,E,R - H-KT_AES_GCM: G,R - H-KT_3DES: G,E,R - H-KT_HMAC: E,R - H-KT_ECDSA_PAIR: E - H-KT_ECDSA_sk: E - H-sEC_KAS_sk: E - E-eEC_KAS_pk: W,E,Z - H-KT_SecretData: G,R - H-Zd: G,E,R,Z - H-eWrap: G,E,Z Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G,E,R - H-KT_AES_CMAC: G,E,R - H-KT_AES_WRAP: G,E,R - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KT_AES_GCM: G,R - H-KT_3DES: G,E,R - H-KT_HMAC: E,R - H-KT_ECDSA_PAIR: E - H-KT_ECDSA_sk: E - H-sEC_KAS_sk: E - E-eEC_KAS_pk: W,E,Z - H-KT_SecretData: G,R - H-Zd: G,E,R,Z - H-eWrap: G,E,Z
Generate Key	Create a new key with user's input algorithm Key value randomized by FIPS applet	Completion Status	Key token properties	Generated key in key token form protected by CCMK of the module as its protection key Return Status	AES DRBG ECDSA KeyGen ECDSA KeyVer KDF RSA KeyGen CKG Entropy source	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_RSA_CRT_PAIR: G - H-KT_RSA_SFM_PAIR: G - H-KT_ECDSA_PAIR: G - H-KT_RSA_OAEP_PAIR: G - H-sEC_KAS_pk:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G - H-sEC_KAS_sk: G - OS-DRBG-SEED: E Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_RSA_CRT_PAIR: G - H-KT_RSA_SFM_PAIR: G - H-KT_ECDSA_PAIR: G - H-KT_RSA_OAEP_PAIR: G - H-sEC_KAS_pk: G - H-sEC_KAS_sk: G - OS-DRBG-SEED: E
Generate Key Pair	Create a new asymmetric key Users can freely choose the algorithm and curve Key value	Completion Status	Key token properties	Generated key in key token form protected by CCMK of the module as protection key	AES KDF	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G - H-KT_AES_CMAC: G - H-KT_AES_WRA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	randomized by FIPS applet			Return Status		P: G - H-KT_AES_GCM: G - H-KT_3DES: G - H-KT_HMAC: G Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G - H-KT_AES_CMAC: G - H-KT_AES_WRAP: G - H-KT_AES_GCM: G - H-KT_3DES: G - H-KT_HMAC: G
Generate Key (For Split Knowledge)	Customer key generation This service will determine the specification of key token output from the whole Customer Key	Completion Status	Customer key properties	Return Status	KDF	Administrator – CO - H-KT_AES: G - H-KT_AES_CMAC: G - H-KT_AES_WRAP: G - H-KT_AES_GCM: G - H-KT_3DES: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	Ceremony procedure					- H-KT_HMAC: G
CK Export 1	Export a secret for customer key creation for KCP1	Completion Status	Only input frame with no additional input data	Customer key's secret data, its KCV and properties Return Status	None	Administrator – CO - H-KT_AES: E Standard User – UR - H-KT_AES: E
CK Export 2	Export a secret for customer key creation for KCP2	Completion Status	Only input frame with no additional input data	Customer key's secret data and its KCV Return Status	None	Administrator – CO - H-KT_AES: E Standard User – UR - H-KT_AES: E
CK Export 3	Export a secret for customer key creation for KCP3 This sequence will return KCV of Secret and Key Token for CK Import procedure later on	Completion Status	Only input frame with no additional input data	Exported Customer key's properties, secret data, secret KCV, customer key KCV Return Status	None	Administrator – CO - H-KT_AES: E - H-CCMKMac: E Standard User – UR - H-KT_AES: E - H-CCMKMac: E
CK Import 1	Import secret based on user's (KCP1) input	Completion Status	Customer key's properties, secret data, and its KCV to import Customer Key using split knowledge procedure	Return Status	None	Administrator – CO Standard User – UR

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
CK Import 2	Import secret based on user's (KCP2) input	Completion Status	Customer key's secret data and its KCV to import Customer Key using split knowledge procedure	Return Status	None	Administrator – CO Standard User – UR
CK Import 3	Import secret and key token based on user's (KCP3) input. This sequence will return a completed key token to be stored in LKD or LKD_DP	Completion Status	Imported Customer key's properties, secret data, secret KCV, and customer key KCV to import Customer Key using split knowledge procedure	Customer Key Token protected by CCMK of the module as its protection key Return Status	AES KDF	Administrator – CO - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G - H-KT_AES_CMAC: G - H-KT_AES_WRAP: G - H-KT_AES_GCM: G - H-KT_3DES: G - H-KT_HMAC: G Standard User – UR - H-CCMKEnc: E - H-CCMKMac: E - H-KT_AES: G - H-KT_AES_CMAC: G - H-KT_AES_WRAP: G - H-KT_AES_GCM: G - H-KT_3DES:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G - H-KT_HMAC: G
Set Date	Set Date in timestamp managet within HSM	Completion Status	Date information	Return Status	None	Super User – SU Administrator – CO
System Reset	Reset admin user, applet life cycle, LKD, and LKD_DP	Completion Status	Only input frame with no additional input data	Return Status	None	Super User – SU - H-CCMK: Z - H-CCMKEnc: Z - H-CCMKMac: Z - H-User_pwd: Z - H-KT_ECDSA_PAIR: Z - H-KT_ECDSA_sk: Z - H-KT_AES: Z - H-KT_AES_CMAC: Z - H-KT_AES_WRAP: Z - H-KT_AES_GCM: Z - H-KT_3DES: Z - H-KT_RSA_CRT_PAIR: Z - H-KT_RSA_SFM_PAIR: Z - H-KT_RSA_CRT_sk: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_RSA_SFM_sk: Z - H-KT_HMAC: Z - H-KT_ECDSA_pk: Z - H-sEC_KAS_pk: Z - H-sEC_KAS_sk: Z - H-KT_RSA_pk: Z - H-sUSER_pk: Z - H-KT_RSA_OAEP_PAIR: Z - H-KT_RSA_OAEP_sk: Z - H-KT_RSA_OAEP_pk: Z - E-sEC_KAS_pk: Z - H-Zd: Z Administrator – CO - H-CCMK: Z - H-CCMKEnc: Z - H-CCMKMac: Z - H-User_pwd: Z - H-KT_ECDSA_PAIR: Z - H-KT_ECDSA_s

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						k: Z - H-KT_AES: Z - H-KT_AES_CMAC: Z - H-KT_AES_WRAP: Z - H-KT_AES_GCM: Z - H-KT_3DES: Z - H-KT_RSA_CRT_PAIR: Z - H-KT_RSA_SFM_PAIR: Z - H-KT_RSA_CRT_sk: Z - H-KT_RSA_SFM_sk: Z - H-KT_HMAC: Z - H-KT_ECDSA_pk: Z - H-sEC_KAS_pk: Z - H-sEC_KAS_sk: Z - H-KT_RSA_pk: Z - H-sUSER_pk: Z - H-KT_RSA_OAEP_PAIR: Z - H-KT_RSA_OAE

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						P_sk: Z - H-KT_RSA_OAEP_pk: Z - E-sEC_KAS_pk: Z - H-Zd: Z
Get Data	Get public key data of super user or host during personalization of the HSM	Completion Status	Requested data such as: System, Host or Super User's Ephemeral public key	Requested data Return Status	None	Super User – SU Administrator – CO
System Store Data	This service is normally used to store SUPER_USER public key within FIPS Applet Initialization procedure after SUPER_USER changes its smartcard's PIN	Completion Status	Data to be stored such as: : System, Host or Super User's Authentication public key, System Authentication private key, System Sync Master Key, logging configuration	Return Status	KAS SSC	Super User – SU - H-sHOST_pk: W - H-sUSER_pk: W Administrator – CO - H-sHOST_pk: W - H-sUSER_pk: W
Sync Request Token	This service is targeted to SE_MASTER to retrieve a sync token	Completion Status	Sync properties	Sync token data Return Status	AES MAC KDA	Super User – SU - H-KSync: E - H-SKSyncEnc: G,E,Z - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	which will be referred on Sync Write Token service					SKSyncMac: G,E,Z Administrator – CO - H-KSync: E - H-SKSyncEnc: G,E,Z - H-SKSyncMac: G,E,Z
Sync Write Token	This service will write sync token on SE_SLAVE. Sync token that is written in this service can be retrieved from Sync Request Token service	Completion Status	Sync token data	Sync result status	AES MAC KDA	Super User – SU - H-KSync: E - H-SKSyncEnc: G,E,Z - H-SKSyncMac: G,E,Z - H-KT_AES: Z - H-KT_AES_CMAC: Z - H-KT_AES_GCM: Z - H-KT_AES_WRAP: Z - H-KT_3DES: Z - H-KT_HMAC: Z - H-KT_RSA_CRT_PAIR: Z - H-KT_RSA_SFM_PAIR: Z - H-KT_RSA_pk: Z - H-KT_RSA_CRT_sk: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_RSA_SFM_sk: Z - H-KT_ECDSA_PAIR: Z - H-KT_ECDSA_pk: Z - H-KT_ECDSA_sk: Z - H-KT_RSA_OAEP_PAIR: Z - H-KT_RSA_OAEP_sk: Z - H-KT_RSA_OAEP_pk: Z - H-sEC_KAS_pk: Z - H-sEC_KAS_sk: Z - E-sEC_KAS_pk: Z - H-Zd: Z Administrator – CO - H-KSync: E - H-SKSyncEnc: G,E,Z - H-SKSyncMac: G,E,Z - H-KT_AES: Z - H-KT_AES_CMAC: Z - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KT_AES_GCM: Z - H-KT_AES_WRAP: Z - H-KT_3DES: Z - H-KT_HMAC: Z - H-KT_RSA_CRT_PAIR: Z - H-KT_RSA_SFM_PAIR: Z - H-KT_RSA_pk: Z - H-KT_RSA_CRT_sk: Z - H-KT_RSA_SFM_sk: Z - H-KT_ECDSA_PAIR: Z - H-KT_ECDSA_pk: Z - H-KT_ECDSA_sk: Z - H-KT_RSA_OAEP_PAIR: Z - H-KT_RSA_OAEP_sk: Z - H-KT_RSA_OAEP_pk: Z - H-sEC_KAS_pk: Z - H-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						sEC_KAS_sk: Z - E-sEC_KAS_pk: Z - H-Zd: Z
Sync Report	This service purpose is to confirm that Sync Write Token had performed successfully. The target of this service is SE_MASTER	Completion Status	Sync result data	Return Status	AES MAC	Super User – SU - H-SKSyncEnc: E,Z - H-SKSyncMac: E,Z Administrator – CO - H-SKSyncEnc: E,Z - H-SKSyncMac: E,Z
Crypto Encipher	Perform encipher on the user's input data	Completion Status	Encipher Key label, plain data and cipher configuration	Encrypted data Return Status	AES AES GCM	Administrator – CO - H-KT_AES: E - H-KT_AES_GCM: E - H-GCM_IV: E Standard User – UR - H-KT_AES: E - H-KT_AES_GCM: E - H-GCM_IV: E
Crypto Decipher	Perform decipher on the user's input data	Completion Status	Decipher Key label, encrypted data and decipher configuration	Decrypted data Return Status	AES AES GCM	Administrator – CO - H-KT_AES: E - H-KT_AES_GCM: E - H-GCM_IV: E Standard User – UR

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- H-KT_AES: E - H-KT_AES_GCM: E - H-GCM_IV: E
Crypto Sign	Create signature based on the user's input data	Completion Status	Signature Key label, data and signature configuration	Signature of the data Return Status	MAC ECDSA SigGen RSA SigGen	Administrator – CO - H-KT_HMAC: E - H-KT_RSA_CRT_PAIR: E - H-KT_RSA_SFM_PAIR: E - H-KT_RSA_CRT_sk: E - H-KT_RSA_SFM_sk: E - H-KT_ECDSA_PAIR: E - H-KT_ECDSA_sk: E - H-KT_AES_CMAC: E Standard User – UR - H-KT_HMAC: E - H-KT_RSA_CRT_PAIR: E - H-KT_RSA_SFM_PAIR: E - H-KT_RSA_CRT_sk: E - H-KT_RSA_SFM

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						_sk: E - H-KT_ECDSA_PAIR: E - H-KT_ECDSA_sk: E - H-KT_AES_CMAC: E
Crypto Verify	Confirm the verified status of the user's signature data	Completion Status	Signature Key label, data, signature of the data, and verification configuration	Signature verification result Return Status	MAC ECDSA SigVer RSA SigVer	Administrator – CO - H-KT_HMAC: E - H-KT_RSA_CRT_PAIR: E - H-KT_RSA_SFM_PAIR: E - H-KT_RSA_pk: E - H-KT_ECDSA_PAIR: E - H-KT_ECDSA_pk: E - H-KT_AES_CMAC: E Standard User – UR - H-KT_HMAC: E - H-KT_RSA_CRT_PAIR: E - H-KT_RSA_SFM_PAIR: E - H-KT_RSA_pk: E - H-KT_ECDSA_P

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						AIR: E - H-KT_ECDSA_pk: E - H-KT_AES_CMAC: E
Crypto Hash	Performs a hash operation on the user's input data	Completion Status	Data, algorithm for hash operation	The resulting hash value Return Status (indicates the success or failure of the operation)	SHA	Administrator – CO Standard User – UR
User Unblock	Unblock a blocked user	Completion Status	Username information	Return Status	None	Administrator – CO
Admin Unblock	unblock Admin	Completion Status	Admin username information	Return Status	None	Super User – SU
System Unblock	unblock HSM	Completion Status	Only input frame with no additional input data	Return Status	None	Super User – SU Administrator – CO
Self-test	On demand self-test execution on rebooting the module	Completion Status	Power cycle	Return Status	None	Unauthenticated

Table 13: Approved Services

4.4 Non-Approved Services

The module does not implement non-approved service.

4.5 External Software/Firmware Loaded

The module employs GP APDU Commands i.e. Load command and Install command as specified in [GPC_Specification_v2.3] as part of GP Manage Content Services for firmware loading. Due to I/O buffer size of the module, the firmware loading process utilizes several Load commands and Install command. Data output is inhibited in each APDU command execution and during Load Test.

The module performs Load Test specified in the section 10.2 Conditional Self-Tests during firmware loading process. If the load test is failed, specific Status Word of the command is returned to indicate failure on firmware loading and the firmware cannot be used.

New firmware version within the scope of this validation must be validated through FIPS 140-3 CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires separate FIPS 140-3 validation. A procedure for firmware loading is described in [FQR 401 9097 Ed 9] section 3.4.2.

JPatch is a standardized procedure employed to apply updates to the firmware authorized by role ‘AA’. It ensures that firmware modifications are implemented reliably, minimizing system down time and preserving device integrity.

4.6 Additional Information

Quorum-based access control is a multi-party access control mechanism implemented within the FIPS Applet, leveraging both identity-based authentication methods: Password-based Authentication, Smartcard-based Authentication. This mechanism requires that K out N authorized users are successfully authenticated to perform sensitive operations or transition into privileged states. The authentication process is governed by the Quorum Profile Definition, which specifies: the total number of eligible users (N), the minimum number of required authentication (K), the roles and credentials associated with each user.

5 Software/Firmware Security

5.1 Integrity Techniques

Initial firmware is loaded through a chip loader (referenced as Initial Flash Loader) and provided by Hardware Manufacturer. The loader is used to decrypt a RookySE firmware delivered in an encrypted firmware. Once full firmware is received and deciphered, a final checksum is sent to the Flash Loader to compare against the internal computed checksum.

From the RookySE firmware, OS checksum can be verified through a GP GET DATA command on DGI DF6E. During this command, the checksum is recalculated on the OS memory range. In case of memory corruption, the card might trigger a security event. OS checksum is checked using a 16-bit EDC and compared against a stored value in NVM.

Beside OS Firmware verification, the module also ensures the Applet Packages integrity by computing a CRC16 on each package and comparing against stored values. The laters are computed during application

loading and stored as references. Upon a failed checksum verification, the OS Firmware will trigger a security event.

5.2 Initiate on Demand

The module permits operators to initiate pre-operational self-tests on demand by power cycling the module.

The operator can perform a warm or cold reset to check OS integrity and NVM (applet packages) integrity is valid or not as specified in section 10.1 Pre-Operational Self-Tests.

5.3 Additional Information

The module consists of several components that have different forms of executable code. The following table shows the form of each component:

Component	Sub Components	Form
OS	Native Application	A binary code written in c and assembly language running on Hardware's CPU
	Built-in Card Manager	A binary code written in Java running on JCVM
FIPS Applet	FIPS Applet	A binary code written in Java running on JCVM
	Core Package	A binary code written in Java running on JCVM
	Rooky Common Package	A binary code written in Java running on JCVM

6 Operational Environment

6.1 Operational Environment Type and Requirements

Not Applicable. The module is designated as a limited operational environment under the FIPS 140-3 definitions. The module includes a firmware load process (GP Manage Content service) to support necessary updates. New firmware versions within the scope of this validation must be validated through FIPS 140-3 CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires separate FIPS 140-3 validation.

Type of Operational Environment: Limited

7 Physical Security

7.1 Mechanisms and Actions Required

The module is a single-chip implementation that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations.

The following table shows the physical security mechanisms that are implemented in the module and the actions required by the operator(s) to ensure that the physical security is maintained:

Mechanism	Inspection Frequency	Inspection Guidance
Tamper-evident coating on chip: The module implements a secure wiring: all security critical wires are protected by special routing measures against probing. Additionally, the wires are embedded into shield lines and used as normal lines for operation to prevent successful probing. Whenever a physical manipulation or physical probing attack is detected, the processing of the module is stopped, and the module enters a secure state (reset)	Permanently active, the detection is automatic	N/A
Memory Protection: All memories present on the module (Flash, ROM, RAM) are encrypted, memory addresses are scrambled and data transferred over bus are masked. Furthermore, RAM, Flash and Cache integrity are protected with error detection mechanisms In case of security critical error, the module enters a secure state (reset).	Permanently active, the detection is automatic	N/A
Sensors: The module is equipped with a temperature sensor, a voltage sensor, a frequency sensor and backside light detection. The module enters a secure state in case of range violation (reset).	Permanently active, the detection is automatic	N/A

Table 14: Mechanisms and Actions Required

7.2 EFP/EFT Information

The following table shows temperature and voltage measurement for EFP that is required for modules with physical Security Level 3.

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	Below -25°C	EFT	Shutdown
HighTemperature	Above +85°C	EFT	Shutdown
LowVoltage	1.40V < Vcc < 1.62V	EFT	Shutdown
HighVoltage	5.5V < Vcc < 7.9V	EFT	Shutdown

Table 15: EFP/EFT Information

7.3 Hardness Testing Temperature Ranges

The following table shows hardness tested at the lowest and highest temperatures within the module's intended temperature range of operation.

Temperature Type	Temperature
LowTemperature	-25°C
HighTemperature	+85°C

Table 16: Hardness Testing Temperatures

8 Non-Invasive Security

Not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The module's SSP storage areas are detailed in the table below.

Storage Area Name	Description	Persistence Type
RAM	Volatile memory	Dynamic
Flash	Non volatile memory	Static

Table 17: Storage Areas

SSPs are securely stored in the Cryptographic Module, a secure element hardware with a tamper active detection and response. The SSPs stored in the module are associated and accessible only via approved services as you can see in this section. The approved services themselves are associated and accessible with some specific roles as described in the table of section 4.3 Approved Services.

9.2 SSP Input-Output Methods

The module's SSP input/output methods are detailed in the table below.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Store Data - Input	External	Flash	Encrypted	Automated	Electronic	GP KTS Unwrap
Diversify Key - Output	RAM	External	Encrypted	Manual	Electronic	KDF
Export CCMK - Output	RAM	External	Encrypted	Manual	Electronic	AES-CMAC (A8099)
Import CCMK – Input	External	Flash	Encrypted	Manual	Electronic	AES-CMAC (A8099)

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
User create – Input	External	Flash	Encrypted	Manual	Electronic	ECDSA KeyVer
User Change Credential – Input	External	Flash	Encrypted	Manual	Electronic	ECDSA KeyVer
Import Key – Input	External	RAM	Encrypted	Automated	Electronic	KAS SSC
Export Key – Output	RAM	External	Encrypted	Automated	Electronic	KAS SSC
Load key – Input	External	RAM	Encrypted	Automated	Electronic	KAS SSC
Mutual Authenticate - Input	External	RAM	Encrypted	Automated	Electronic	KAS ECC
Mutual Authenticate – Output	RAM	External	Encrypted	Automated	Electronic	KAS ECC
Cipher/Decipher – Input	External	RAM	Encrypted	Manual	Electronic	AES
Secure Channel	RAM/Flash	External	Encrypted	Automated	Electronic	GP KTS Wrap

Table 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The module's SSP zeroization methods are detailed in the table below.

Zeroization Method	Description	Rationale	Operator Initiation
Zeroized after used	Overwrite with zero values	Overwrite with zero values	N/A
APDU Command GP Delete Key	Erases the content of OS persistent keys	Overwrite with zero values	By calling the Delete Key command of GP
Applet selection	Erases the content of OS volatile keys	Overwrite with zero values	By calling GP Select commad
Manage session	Destroying SSP related to session when it is closed	Overwrite with zero values	By calling Manage Session service
User logout	Destroying key related to user session	Overwrite with zero values	By calling User Logout service
User deletion	Destroying user's credential as password or public key when a user is	Overwrite with zero values	By calling User Delete service

Zeroization Method	Description	Rationale	Operator Initiation
	deleted from the application		
System Reset	Destroying SSP stored in volatile memory	Overwrite with zero values	By calling System Reset or SE Reset services
Synchronization	Destroying SSP during synchronisation	Overwrite with zero values	By calling Sync Write Token or Sync Request Token services
Module entering TERMINATED state	Destroying the SSPs by overwriting with zero values	Overwrite with zero values	By setting the module in the TERMINATED state, by calling the GP Life Cycle service
Failure	Destroying ephemeral keys before entered to error state	Overwrite with zero values	When an error occurs
Firmware Loading Process using APDU Command Delete Package and Instance	Destroying applet keys during firmware update	Overwrite with zero values	By calling GP Manage Content Service

Table 19: SSP Zeroization Methods

For all SSP, an implicit indicator that the zeroization is completed is provided. This indicator is the successful completion of the requested service for SSP zeroized through a dedicated command (the command is indicated in column "zeroization" of Table 21) or Answer-to-RESET (ATR) in case of module reset. For SSP that are automatically zeroized (e.g H-Zs and H-Ze), the implicit indicator is the completion of the command mentioned in Table 21 with a correct status or an error status.

9.4 SSPs

The module's SSPs are detailed in the tables below.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
OS-DRBG-SEED	Entropy input and nonce provided by the Entropy Source	N/A - N/A	Seed - CSP	Entropy source		DRBG
OS-DRBG-STATE	The current AES-256 CTR_DRBG state composed of values V and Key	N/A - N/A	DRBG state - CSP	DRBG		DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SD-KENC	Master Encryption Security Domain key	256 - 256	Symmetric Key - CSP			AES
SD-KMAC	Master MAC Security Domain key	256 - 256	Symmetric Key - CSP			AES
SD-KDEK	Master DEK Security Domain key	256 - 256	Symmetric Key - CSP			AES
SD-SENC	GP Secure Channel Session Encryption Key	256 - 256	Symmetric Key - CSP		GP KTS Wrap GP KTS Unwrap	GP KTS Wrap GP KTS Unwrap
SD-SMAC	GP Secure Channel Session MAC Key	256 - 256	Symmetric Key - CSP		GP KTS Wrap GP KTS Unwrap	GP KTS Wrap GP KTS Unwrap
SD-RMAC	GP Secure Channel Session Response MAC Key	256 - 256	Symmetric Key - CSP		GP KTS Wrap GP KTS Unwrap	GP KTS Wrap GP KTS Unwrap
DAP-AES	Data Authentication Pattern AES Key	128 - 128	Symmetric Key - CSP		MAC	
JPATCH-AES	JPATCH Master Key (imported during secure OS generation in secure environment - before manufacturing)	256 - 256	Symmetric Key - CSP			AES
JPATCH-AES-SENC	JPATCH Encryption Key	256 - 256	Symmetric Key - CSP			AES
JPATCH-AES-SMAC	JPATCH MAC Key	256 - 256	Symmetric Key - CSP			MAC
H-sAUTH_sk	System Authenticati	P-521 - 256	EC Private			KAS ECC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	on Static Private Key		Key - CSP			
H-eAUTH_sk	System Authentication Ephemeral Private Key	P-521 - 256	EC Private Key - CSP	CKG		KAS ECC
H-Zs	Shared secret Zs	521 - 256	Shared secret - CSP		KAS ECC	
H-Ze	Shared secret Ze	521 - 256	Shared Secret - CSP		KAS ECC	
H-Zd	Ephemeral Shared Secret Zd	224 to 521 - 112 to 256	Shared Secret - CSP		KAS SSC	
H-SKAuthEnc	Encryption Secure Channel Session Key	256 - 256	Symmetric Key - CSP		KAS ECC	AES
H-SKAuthMac	MAC Secure Channel Session Key	256 - 256	Symmetric Key - CSP		KAS ECC	MAC
H-SKAuthKC	Key Confirmation Secure Channel Session Key	256 - 256	Symmetric Key - CSP		KAS ECC	MAC
H-CCMK	Crypto Card Master Key	256 - 256	Symmetric Key - CSP	KDF		
H-CCMKEnc	Encryption Derived CCMK Key	256 - 256	Symmetric Key - CSP	KDF		AES
H-CCMKMac	MAC Derived CCMK Key	256 - 256	Symmetric Key - CSP	KDF		MAC
H-KSync	Synchronization Master Key	256 - 256	Symmetric Key - CSP			KDF
H-SKSyncEnc	Encryption Synchronization Session Key	256 - 256	Symmetric Key - CSP	KDF		AES

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
H-SKSyncMac	MAC Synchronization Session Key	256 - 256	Symmetric Key - CSP	KDF		MAC
H-User_pwd	User's password credential	Configurable with maximum 16 characters - Configurable with maximum 16 characters	Password - CSP			KAS ECC
H-KT_ECDSA_PAIR	Key Token ECDSA PAIR	P-192 to P-521 - 80 to 256	ECC Key Pair - CSP	ECDSA KeyGen		ECDSA KeyGen ECDSA KeyVer ECDSA SigGen ECDSA SigVer
H-KT_ECDSA_sk	Key Token ECDSA PRIVATE	P-224 to P-521 - 112 to 256	EC Private key - CSP	ECDSA KeyGen		ECDSA SigGen
H-eEC_KAS_sk	HSM Ephemeral EC KAS PRIVATE	P-224 to P-521 - 112 to 256	EC Private Key - CSP	KAS SSC		KAS SSC
H-sEC_KAS_sk	HSM Static EC KAS PRIVATE	P-224 to P-521 - 112 to 256	EC Private Key - CSP	CKG		KAS SSC
H-eWrap	Ephemeral Key	128 to 256 - 128 to 256	Symmetric Key - CSP			KDF
H-KT_AES	Key Token AES	128 to 256 - 128 to 256	Symmetric Key - CSP	KDF		AES
H-KT_AES_CMAC	Key Token AES CMAC	128 to 256 - 128 to 256	Symmetric Key - CSP	KDF	KDA	MAC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
H-KT_AES_WRAP	Key Token AES KW/KWP	128 to 256 - 128 to 256	Symmetric Key - CSP	KDF	KDA	AES Wrap AES Unwrap
H-KT_AES_GCM	Key Token AES GCM	128 to 256 - 128 to 256	Symmetric Key - CSP	KDF		AES GCM
H-KT_RSA_CRT_PAIR	Key Token RSA CRT PAIR	1024 to 4096 - 80 to 128	RSA CRT Key Pair - CSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_SFM_PAIR	Key Token RSA SFM PAIR	1024 to 4096 - 80 to 128	RSA SFM Key Pair - CSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_CRT_sk	Key Token RSA CRT PRIVATE	2048 to 4096 - 112 to 128	RSA CRT Private Key - CSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_SFM_sk	Key Token RSA SFM PRIVATE	2048 to 4096 - 112 to 128	RSA SFM Private Key - CSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_OAEP_PAIR	Key Token RSA OAEP PAIR	2048 to 4096 - 112 to 128	RSA OEAP Key Pair - CSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_OAEP_sk	Key Token RSA OAEP PRIVATE	2048 to 4096 - 112 to 128	RSA OEAP Privat Key - CSP	CKG		KTS Encap KTS Decap
H-KT_HMAC	Key Token HMAC. HMAC-64 is only used for MAC verification.	64 to 512 - 64 to 256	HMAC Key - CSP	KDF	KDA	MAC
H-KT_ECDSA_pk	Key Token ECDSA PUBLIC	P-192 to P-521 - 96 to 256	ECC Public Key - PSP	ECDSA KeyGen		ECDSA KeyVer ECDSA SigGen ECDSA SigVer
H-eEC_KAS_pk	HSM Ephemeral	P-244 to P-521 -	EC Public Key - PSP	CKG		KAS SSC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	EC KAS PUBLIC	112 to 256				
H-sEC_KAS_pk	HSM Static EC KAS PUBLIC	P-244 to P-521 - 112 to 256	EC Public Key - CSP	CKG		KAS SSC
E-eEC_KAS_pk	External Ephemeral EC KAS PUBLIC	P-244 to P-521 - 112 to 256	EC Public Key - PSP			KAS SSC
E-sEC_KAS_pk	External Static EC KAS PUBLIC	P-244 to P-521 - 112 to 256	EC Public Key - PSP			KAS SSC
H-KT_RSA_pk	Key Token RSA PUBLIC	1024 to 4096 - 80 to 128	RSA Public Key - PSP	RSA KeyGen		RSA SigGen RSA SigVer
H-KT_RSA_OAEP_pk	Key Token RSA OAEP PUBLIC	2048 to 4096 - 112 to 128	RSA OAEP Public Key - PSP	CKG		KTS Encap KTS Decap
H-sAUTH_pk	System Authentication Static Public Key	P-521 - 256	EC Public Key - PSP			KAS ECC
H-sHOST_pk	HOST Authentication Static Public Key	P-521 - 256	EC Public Key - PSP			KAS ECC
H-sUSER_pk	User Authentication Static Public Key	P-521 - 256	EC Public Key - PSP			KAS ECC
H-eAUTH_pk	System Authentication Ephemeral Public Key	P-521 - 256	EC Public Key - PSP	CKG		KAS ECC
H-eHOST_pk	HOST / SMA Authentication Ephemeral Public Key	P-521 - 256	EC Public Key - PSP			KAS ECC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
H-HsmID	Predefined System Identification data	N/A - N/A	Identification data - PSP			KDF
H-HostID	HOST Identification data	N/A - N/A	Identification data - PSP			KDF
H-GCM_IV	IV of AES GCM	96 - 96	IV - PSP			AES GCM
H-KT_3DES	Key Token DES	64 to 192 - 64 to 192	Other parameters - Neither	KDF	KDA	
H-KT_SecretData	Secret Data	128 to 256 - 128 to 256	Other parameters - Neither		KDA	
OS-MKEK	Key Encryption Key (AES key) - Master Key used to encrypt (obfuscate) storage of CSPs	128 - 128	Other parameters - Neither	KDF		CSPs obfuscation

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
OS-DRBG-SEED		RAM:Plaintext	Duration of the module session	Zeroized after used	
OS-DRBG-STATE		RAM:Plaintext	Duration of the module session	Zeroized after used	
SD-KENC	Store Data - Input	Flash:Obfuscated	Persistent until zeroized	APDU Command GP Delete Key	SD-SENC:Derives
SD-KMAC	Store Data - Input	Flash:Obfuscated	Persistent until zeroized	APDU Command GP Delete Key	SD-SMAC:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SD-KDEK	Store Data - Input	Flash:Obfuscated	Persistent until zeroized	APDU Command GP Delete Key	SD-KENC:Decrypts SD-KMAC:Decrypts SD-KDEK:Decrypts DAP-AES :Decrypts
SD-SENC		RAM:Plaintext	Duration of the secure channel session	Applet selection System Reset	SD-KENC:Derived From
SD-SMAC		RAM:Plaintext	Duration of the secure channel session	Applet selection System Reset	SD-KMAC:Derived From
SD-RMAC		RAM:Plaintext	Duration of the secure channel session	Applet selection System Reset	
DAP-AES	Secure Channel	Flash:Plaintext	Persistent until zeroized	APDU Command GP Delete Key	
JPATCH-AES		Flash:Plaintext	Persistent until zeroized	Module entering TERMINATED state	
JPATCH-AES-SENC		RAM:Plaintext	Duration of the module session	Applet selection System Reset	JPATCH-AES:Derived From
JPATCH-AES-SMAC		RAM:Plaintext	Duration of the module session	Applet selection System Reset	JPATCH-AES:Derived From
H-sAUTH_sk	Store Data - Input Secure Channel	Flash:Plaintext	Persistent until zeroized	Firmware Loading Process using APDU Command Delete Package and Instance	H-sHOST_pk:Used With H-eAUTH_pk:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
H-eAUTH_sk		RAM:Plaintext	During authentication	Zeroized after used System Reset Failure	H-eHOST_pk:Used With
H-Zs		RAM:Plaintext	During authentication	Zeroized after used System Reset Failure	H-Ze:Used With H-SKAuthEnc:Derived From H-SKAuthMac:Derived From H-SKAuthKC:Derived From
H-Ze		RAM:Plaintext	During authentication	Zeroized after used System Reset Failure	H-Zs:Used With H-SKAuthEnc:Derived From H-SKAuthMac:Derived From H-SKAuthKC:Derived From
H-Zd	Diversify Key - Output	RAM:Plaintext	During key establishment	Zeroized after used System Reset Failure	H-eEC_KAS_sk:Derived From E-sEC_KAS_pk:Derived From E-eEC_KAS_pk:Derived From H-sEC_KAS_sk:Derived From
H-SKAuthEnc		RAM:Plaintext	Duration of the user session	Manage session User logout System Reset Failure	
H-SKAuthMac		RAM:Plaintext	Duration of the	Manage session User	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			user session	logout System Reset Failure	
H-SKAuthKC		RAM:Plaintext	Duration of the user session	Manage session User logout System Reset Failure	
H-CCMK	Export CCMK - Output Import CCMK – Input	Flash:Plaintext	Persistent until zeroized	System Reset Firmware Loading Process using APDU Command Delete Package and Instance	
H-CCMKEnc		Flash:Plaintext	Persistent until zeroized	System Reset Firmware Loading Process using APDU Command Delete Package and Instance	H-CCMK:Derived From
H-CCMKMac		Flash:Plaintext	Persistent until zeroized	System Reset Firmware Loading Process using APDU Command Delete Package and Instance	H-CCMK:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
H-KSync	Store Data - Input	Flash:Plaintext	Persistent until zeroized	Firmware Loading Process using APDU Command Delete Package and Instance	
H-SKSyncEnc		RAM:Plaintext	During the module session	Manage session System Reset Failure	H-KSync:Derived From
H-SKSyncMac		RAM:Plaintext	During the module session	Manage session System Reset Failure	H-KSync:Derived From
H-User_pwd	User create – Input User Change Credential – Input	Flash:Obfuscated	Persistent until zeroized	User deletion System Reset Synchronization Firmware Loading Process using APDU Command Delete Package and Instance	
H-KT_ECDSA_PAIR	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_ECDSA_sk:Paired With H-KT_ECDSA_pk:Paired With
H-KT_ECDSA_sk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering	H-KT_ECDSA_PAIR:Paired With H-

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				TERMINATED state	KT_ECDSA_pk:Paired With
H-eEC_KAS_sk		RAM:Plaintext	After successful security services	Zeroized after used System Reset Failure	H-eEC_KAS_sk:Used With E-sEC_KAS_pk:Used With
H-sEC_KAS_sk	Import Key – Input Export Key – Output Load key – Input	RAM:Plaintext	After successful security services	Manage session System Reset Synchronization Module entering TERMINATED state	E-eEC_KAS_pk:Used With H-sEC_KAS_sk:Used With
H-eWrap		RAM:Plaintext	After successful security services	Zeroized after used System Reset Failure	
H-KT_AES	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
H-KT_AES_CMAC	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
H-KT_AES_WRAP	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
H-KT_AES_GCM	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
H-KT_RSA_CRT_PAIR	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_CRT_sk:Paired With H-KT_RSA_pk:Paired With
H-KT_RSA_SFM_PAIR	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_SFM_sk:Paired With H-KT_RSA_pk:Paired With
H-KT_RSA_CRT_sk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_CRT_PAIR:Paired With H-KT_RSA_pk:Paired With
H-KT_RSA_SFM_sk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_SFM_sk:Paired With H-KT_RSA_pk:Paired With
H-KT_RSA_OAEP_PAIR	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_OAEP_sk:Paired With H-KT_RSA_OAEP_pk:Paired With
H-KT_RSA_OAEP_sk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_OAEP_PAIR:Paired With H-KT_RSA_OAEP_pk:Paired With
H-KT_HMAC	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
H-KT_ECDSA_pk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering	H-KT_ECDSA_PAIR:Paired With H-

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				TERMINATED state	KT_ECDSA_sk:Paired With
H-eEC_KAS_pk	Import Key – Input Export Key – Output	RAM:Plaintext	After successful security services	Zeroized after used System Reset Failure	H-eEC_KAS_sk:Paired With
H-sEC_KAS_pk	Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-sEC_KAS_sk:Paired With
E-eEC_KAS_pk	Import Key – Input Load key – Input	RAM:Plaintext	After successful security services	Zeroized after used Failure	H-sEC_KAS_sk:Used With
E-sEC_KAS_pk	Load key – Input	RAM:Plaintext	After successful security services	Zeroized after used Failure	H-eEC_KAS_sk:Used With
H-KT_RSA_pk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_CRT_PAIR:Paired With H-KT_RSA_SFM_PAIR:Paired With H-KT_RSA_CRT_sk:Paired With H-KT_RSA_SFM_sk:Paired With
H-KT_RSA_OAEP_pk	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	H-KT_RSA_OAEP_sk:Paired With H-KT_ECDSA_PAIR:Paired With
H-sAUTH_pk	Store Data - Input Secure Channel	Flash:Obfuscated	Persistent until zeroized	Firmware Loading Process using APDU Command	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Delete Package and Instance	
H-sHOST_pk	Store Data - Input	Flash:Plaintext	Persistent until zeroized	Firmware Loading Process using APDU Command Delete Package and Instance	H-sAUTH_sk:Used With
H-sUSER_pk	User create – Input User Change Credential – Input Secure Channel	Flash:Obfuscated	Persistent until zeroized	User deletion System Reset Module entering TERMINATED state Firmware Loading Process using APDU Command Delete Package and Instance	H-sAUTH_sk:Used With
H-eAUTH_pk	Mutual Authenticate – Output	RAM:Plaintext	After key establishment	Zeroized after used System Reset Failure	
H-eHOST_pk	Mutual Authenticate - Input	RAM:Plaintext	After key establishment	Zeroized after used System Reset Failure	H-eAUTH_sk:Used With
H-HsmID		Flash:Plaintext	Persistent until zeroized	Firmware Loading Process using APDU	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Command Delete Package and Instance	
H-HostID	Mutual Authenticate - Input	RAM:Plaintext	During the module session	System Reset	
H-GCM_IV	Cipher/Decipher – Input	RAM:Plaintext	After successful security services	Zeroized after used System Reset Failure	H-KT_AES_GCM:Used With
H-KT_3DES	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
H-KT_SecretData	Import Key – Input Export Key – Output	RAM:Plaintext	During the module session	System Reset Module entering TERMINATED state	
OS-MKEK		Flash:Plaintext	Persistent until zeroized	Module entering TERMINATED state	

Table 21: SSP Table 2

9.5 Transitions

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

Starting January 1, 2031, the minimum security strength for applying cryptographic protection will be 128 bits, and security strength between 112 and 128 bits will be allowed for legacy use only to process information that is already protected.

10 Self-Tests

10.1 Pre-Operational Self-Tests

The module implements the pre-operational self-tests described in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
CRC-16	16-bit EDC		Critical Function	Fail: Module in KillCard State	Computes CRC-16 from a fixed message and checks the result
NVM Integrity		Integrity test	SW/FW Integrity	Fail: Module in KillCard State	CRC-16 performed over all executable (Java Card packages) in NVM
ROM Code Integrity		Integrity test	SW/FW Integrity	Fail: Module in KillCard State	CRC-16 performed over all ROM code

Table 22: Pre-Operational Self-Tests

If any pre-operational self-test fails, the module will enter in the Kill Card state and emit an error code that identifies the type of test that failed. No further communication with the module is possible until the module is reset (Power-On). If it happens several times and reaches the error limit, the module will be terminated.

10.2 Conditional Self-Tests

The module implements the conditional self-tests described in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
CRC-16	CRC-16 bit	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	CRC	Power Up
AES-ECB (A8099)	AES-128-ECB	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Decrypt	Power Up
Counter DRBG Instanciate (A8099)	CTR_DRBGAES 256	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Instanciate	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG Generate (A8099)	CTR_DRBGAES 256	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
Counter DRBG Reseed (A8099)	CTR_DRBGAES 256	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Reseed	Power Up
HMAC-SHA-1 (A8099)	HMAC SHA-1 using 128-bit key	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
SHA2-256 (A8099)	SHA2-256 – bit	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
SHA2-512 (A8099)	SHA2-512-bit	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
SHA3-512 (A8099)	SHA3-512-bit	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
KDF SP800-108 (A8099)	KDF based on AES-128-CMAC	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Generate	Power Up
ECDSA SigGen (FIPS186-5) (A8099)	P-224 curve	KAT	CAST	Pass: Next test - Fail: Module in	Signature generation	First Use

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				KillCard State		
ECDSA SigVer (FIPS186-5) (A8099)	P-224 curve	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Signature verification	First Use
RSA SigGen (FIPS186-5) (A8099)	2048-bit RSA PSS with SHA2-256, using both STD and CRT key formats	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Signature generation	First Use
RSA SigVer (FIPS186-5) (A8099)	2048-bit RSA PSS with SHA2-256	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Signature verification	First Use
AES-CTR Encrypt (A8099)	AES-128-bit-CTR	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Encrypt	Power Up
AES-CTR Decrypt (A8099)	AES-128-bit-CTR	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Decrypt	Power Up
AES-GCM Encryption (A8099)	AES-128-GCM	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Encrypt	Power Up
AES-GCM Decryption (A8099)	AES-128-GCM	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Decrypt	Power Up
AES-KW Wrap (A8099)	AES-128-KW	KAT	CAST	Pass: Next test - Fail:	Wrap	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Module in KillCard State		
AES-KW Unwrap (A8099)	AES-128-KW	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Unwrap	Power Up
KTS-IFC Encryption (A8099)	2048-bit RSA-OAEP with SHA2-256 using both STD and CRT key formats	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Encryption	Power Up
KTS-IFC Decryption (A8099)	2048-bit RSA-OAEP with SHA2-256 using both STD and CRT key formats	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	Decryption	Power Up
KAS-ECC-SSC Sp800-56Ar3 (A8099)	P-224 curve	KAT	CAST	Pass: Next test - Fail: Module in KillCard State	EC CDH for SSC	Power Up
KDA OneStep SP800-56Cr2 (A8099)	One-step KDA based on SHA2-224	KAT	CAST	Pass: Next test - Fail: Application error State	One Step Key Derivation	Power Up
ECDSA KeyGen (FIPS186-5) (A8099)	Approved curve with SHA2-256	ECDSA Pairwise Consistency Test (PCT)	PCT	Pass: Next execution - Fail: Return error code	ECDSA	Each key generation based on key usage SIGN-VERIF, TRANSPORT, DERIVATION
RSA KeyGen (FIPS186-5) (A8099)	Approved RSA Key length	RSA Pairwise Consistency Test (PCT)	PCT	Pass: Next execution - Fail: Return error code	RSA	Each key generation based on key usage sign verify

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
DAP Verification	AES-128-CMAC	Firmware Load Test	SW/FW Load	Pass: Next execution - Fail: Return SW 6985	AES CMAC	While doing the firmware upload test
JPATCH MAC Verification	AES-256-CMAC	Firmware Load Test	SW/FW Load	Pass: Next execution - Fail: Return SW 6985	AES CMAC	While doing the firmware upload test
TRNG	Transition Count health test	Fault Detection Test	CAST	Fail: Module in KillCard State	Performs the start-up and continuous health testing of the entropy source	Power up and before the use of each output of entropy source

Table 23: Conditional Self-Tests

If any self-test fails other than the pairwise consistency, manual entry test, and firmware load test in the conditional self-test, the module will enter in the Kill Card state and emit an error code that identifies the type of test that failed. No further communication with the module is possible until the module is reset (Power-On). If it happens several times and reaches the error limit, the module will be terminated.

The module employs Known Answer Test (KAT) to perform Cryptographic Algorithm Self-test. For performance concern, not all algorithms are executed in the startup (power on) but prior to its first use.

Conditional Software/Firmware Load Test
The module relies on DAP Verification specified in [GPC_Specification_v2.3] section 9.2.1. The employed DAP Verification for Load Test is using AES-CMAC as approved data authentication technique to verify the validity of the firmware that is loaded. The AES Key with 128 bits key length (DAP-AES) is loaded at manufacturing stage and used as authentication key to calculate the MAC of the loaded firmware.

The loading of a JPATCH is identical to the Applet loading. The DAP verification is applied to the authentication of the JPATCH. The MAC verification is applied too to the authentication of the JPATCH using Mac Derivate JPSK. A Software/Firmware Load test is valid for both Java Card packages (Applet, JPatch).

Conditional Pair-Wise Consistency Test
When the module generates RSA and ECC Key Pair via 'Generate Key Pair' service, the module performs pairwise consistency test using signs and verify of known value technique. If pairwise consistency test is

failed, the module returns error code value 'D6' indicating error in Generate Key Service with reason pair-wise consistency self-test is failed. With five consecutive errors, the module will enter blocked state. An exit procedure from this state is described in [FQR 401 9097 Ed 9] section 7.3.

Conditional Manual Entry Test

The module performs manual entry self-test on some services such as "CCMK Import" services done by each key custodian to enter each secret of CCMK and "CK Import" services done by each customer key custodian enter secrets of customer key. Those operators enter KCV along with its secret. The module performs manual entry self-test by comparing entered KCV and calculated KCV of given secret. If the entered KCV does not match calculated KCV, the manual entry test fails with error code 'D7'. On five consecutive errors, the module enters blocked state. An exit procedure from this state is described in [FQR 401 9097 Ed 9] section 7.3.

10.3 Periodic Self-Test Information

The module provides a periodic self-test that is executed in every certain number of FIPS Applet service execution. This number is configured on FIPS Applet installation with 2,097,151 as default value. This periodic self-test executes the self-tests described in section 10.1 Pre-Operational Self-Tests. Data output is inhibited during self-test execution. Only power reset, Hard-fault, procedure NULL ('60') byte in ISO7816 interface, and check alive command in SPI can interrupt the process. If it's interrupted by the Hard fault, the module enters error state of the hardware and requires power reset to exit from this state.

The module's periodic self-tests are detailed in the tables below.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
CRC-16		Critical Function	2.097.151	On Demand & Auto
NVM Integrity	Integrity test	SW/FW Integrity	2.097.151	On Demand & Auto
ROM Code Integrity	Integrity test	SW/FW Integrity	2.097.151	On Demand & Auto

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
CRC-16	KAT	CAST	2.097.151	On Demand & Auto
AES-ECB (A8099)	KAT	CAST	2.097.151	On Demand & Auto
Counter DRBG Instanciate (A8099)	KAT	CAST	2.097.151	On Demand & Auto
Counter DRBG Generate (A8099)	KAT	CAST	2.097.151	On Demand & Auto

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG Reseed (A8099)	KAT	CAST	2.097.151	On Demand & Auto
HMAC-SHA-1 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
SHA2-256 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
SHA2-512 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
SHA3-512 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
KDF SP800-108 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
ECDSA SigGen (FIPS186-5) (A8099)	KAT	CAST	2.097.151	On Demand & Auto
ECDSA SigVer (FIPS186-5) (A8099)	KAT	CAST	2.097.151	On Demand & Auto
RSA SigGen (FIPS186-5) (A8099)	KAT	CAST	2.097.151	On Demand & Auto
RSA SigVer (FIPS186-5) (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-CTR Encrypt (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-CTR Decrypt (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-GCM Encryption (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-GCM Decryption (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-KW Wrap (A8099)	KAT	CAST	2.097.151	On Demand & Auto
AES-KW Unwrap (A8099)	KAT	CAST	2.097.151	On Demand & Auto
KTS-IFC Encryption (A8099)	KAT	CAST	2.097.151	On Demand & Auto
KTS-IFC Decryption (A8099)	KAT	CAST	2.097.151	On Demand & Auto

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KAS-ECC-SSC Sp800-56Ar3 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
KDA OneStep SP800-56Cr2 (A8099)	KAT	CAST	2.097.151	On Demand & Auto
ECDSA KeyGen (FIPS186-5) (A8099)	ECDSA Pairwise Consistency Test (PCT)	PCT		
RSA KeyGen (FIPS186-5) (A8099)	RSA Pairwise Consistency Test (PCT)	PCT		
DAP Verification	Firmware Load Test	SW/FW Load		
JPATCH MAC Verification	Firmware Load Test	SW/FW Load		
TRNG	Fault Detection Test	CAST	2.097.151	On Demand & Auto

Table 25: Conditional Periodic Information

10.4 Error States

The module's self-test error states are described in the table below.

Name	Description	Conditions	Recovery Method	Indicator
KillCard State (Low-level Error State)	A state that indicates the module is in security state. Module at this state can no longer communicate till a power reset (cold or warm) is performed	Pre-Operational Self-Tests Failure, Conditional Cryptographic Algorithm Test Failure, Periodic Self-Test Failure	Power cycle	Muted Device
Application Error State	A state that indicates the module service execution is failed with specific error code. After returning the error code, the module is automatically back to its previous state.	Conditional Cryptographic Algorithm Test Failure	No recovery method is required unless it occurs several times. In that case, HSM unblock service is required.	The module returns error code '0xBE'. Module life cycle can be set to blocked state, it occurs several times.

Table 26: Error States

10.5 Operator Initiation of Self-Tests

The module permits operators to initiate pre-operational self-tests on demand by power cycling the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is delivered as single chip to the customer where its life cycle state is in SYSTEM INITIALIAZATION state. The customer shall follow the procedures below for secure installation, initialization, startup and operation of the module:

1. The installation procedure of the module that supports two physical interfaces, ISO7816 T0 protocol and SPI Protocol are described as follows:
   a. Each ISO7816 physical port defined in section 3.1 Ports and Interfaces must be installed correctly by connecting each ISO port (VCC, GND, RST, CLK, IO) to the module to the corresponding ISO Reader's port or terminal's port. Electrical characteristics in each connected port, signal sequence in activation and deactivation, and transmission protocol shall follow [ISO/IEC 7816-3].
   b. Each SPI physical port defined in section 3.1 Ports and Interfaces must be installed correctly by connecting each SPI port (VCC, GND, RST, SPI_CLK, SPI_MISO, SPI_MOSI, SPI_CS) of the module to the corresponding SPI Reader's port or SPI master device's port. Electrical characteristics in each connected port and its transmission protocol shall follow chip manufacturer datasheet.
2. All module components specified in section 2.2 Tested and Vendor Affirmed Module Version and Identification are already installed in manufacturing. The configuration is set to FIPS Certified Product (FCP).
3. In the SYSTEM INITIALIZATION state:
   a. Super User shall create new credentials and replace its default credential in the module.
      i. In Quorum Super User Mode, a designated Super User is authorized to redefine the PROFILE SUPER USER configuration. This includes the capability to create up to N−1 additional Super User accounts, ensuring that the total number of Super Users remains within the defined quorum limits.
   b. Then, Super User shall create new Administrator role in the module.
      ii. In Quorum Super User Mode, the following administrative operations are permitted:
         1) K out of N Super User must successfully authenticate to initiative privileged operations.
         2) and define the Profile Admin.
         3) Create N Admin users.

When this procedure is successfully done, the module state is automatically changed to KEY CEREMONY state.

4. In the KEY CEREMONY state:
   a. Admin authentication. In Quorum Mode, administrative access requires K out of N Admins authentication.
   b. Administrator shall create new roles for Auditor and three Key Custodians role.
   c. Then, Administrator shall initiate KEY CEREMONY session together with three Key Custodians to import or export CCMK of the module using split knowledge procedure.

d. After that Administrator shall confirm whether to keep the configuration still in FIPS Certified Product (FCP) or not. Please note that if Administrator is answering no, then the module will lose its FIPS Certified status. Customers cannot revert to FIPS Certified status unless the module is returned at IDEMIA to be erased and personalized with new SSPs. It is recommended for customers to perform System Reset before returning the module to the IDEMIA.

When this procedure is successfully done, the module state is automatically changed to System USER (operational) state.

5. When the module is in operational state and configured as FIPS Certified Product (FCP):
   a. Only approved mode of operation is available for users. In this mode of operation, only approved services in section 4.3 Approved Services are available for users.
   b. Only applets that are already validated under [NIST.FIPS.140-3] evaluation shall be loaded and installed in the module. Otherwise, the module will lose its FIPS certified status. This is not automatically enforced by the module but must be obeyed by following procedure defined in [FQR 401 9097 Ed 9] section 3.4.2.
6. The module does not support maintenance role.
7. The Administrator shall update current date of the module. It is strongly recommended to update the current date of the module on daily basis.
8. Detailed information about module life cycle state and procedures for initialization and operation of the module at customer side, and the procedure to keep the module still in FIPS are described in documents [FQR 401 9097 Ed 9] for Crypto Officer Role and [FQR 401 9098 Ed 9] for User Role.

11.2 Administrator Guidance

For detailed administrator guidance, please refer to FQR 401 9097 Ed 9 - IDEMIA Sphere HSM Crypto Officer Guidance.

11.3 Non-Administrator Guidance

For detailed administrator guidance, please refer to FQR 401 9098 Ed 9 - IDEMIA Sphere HSM User.

11.4 End of Life

Sanitization can be done by performing HSM Reset that is authorized by SUPER_USER and HSM Administrator credentials authenticated in Admin Session. HSM Reset service will perform zeroization that is created at customer side and set back module to Factory State like when the customer receives the module for the first time.

For secure destruction procedure, it is recommended for customers to follow these steps:

1. Under role ‘’AA” (Application Administrator), set the card manager state to TERMINATED.
2. Once it is terminated, OS-MKEK (which encrypts all keys stored in the module) is zeroized.
3. Once OS-MKEK is zeroized, all keys stored within the module cannot be retrieved in plaintext form. Even in encrypted form, recovering those keys is extremely difficult, as they are stored in flash memory on the single-chip component protected by a tamper-evident coating.

11.5 Additional Information

Hardware version can be retrieved with the following steps:

- The current selected application shall be the CARD MANAGER
  APDU command: 00 A4 04 00 08 A0 00 00 01 51 00 00 00
- GP GET DATA command will return the hardware version under two possible DGI TAG.
  - DGI 'DF50'
    APDU command: 80 CA DF 50 1B
    Hardware version is in the first byte of IC Type information in the returned data
    Version number: '29'
  - DGI 'DF52'
    APDU command: 80 CA DF 52
    Hardware version is the first byte of the returned data
    Version number: '29'
- The hardware version returned from GET DATA command is act as the hardware identifier and version at the same time

OS Firmware information can be retrieved with the following steps:

- The current selected application shall be the CARD MANAGER
  APDU command: 00 A4 04 00 08 A0 00 00 01 51 00 00 00
- GP GET DATA command will return the OS firmware information under the following information:
  - DGI 'DF66'
    APDU command: 80 CA DF 66 0D
    Returned data is the OS Firmware Information
- OS Firmware Information retrieved by the GET DATA command contains the module identifier and version number, also JPatch update version. The details are as follows:
  - Complete OS Firmware Information: '097154'
    Module Identifier: '09715'
    Version number: '4'
  - JPatch update version (4 bytes)

Application Firmware FIPS Applet version number can be retrieved with the following steps:

- The current selected application shall be the FIPS APPLET
  APDU command: 00 A4 04 00 08 00 00 00 00 00 A1 12 58
- FRAME HSM GET INFO with tag option 'FF' will return the FIPS APPLET version number under tag 'C0'

12 Mitigation of Other Attacks

12.1 Attack List

The Module implements defenses against:

- Fault attacks: the chip includes several sensors for detecting intrusion or fault attacks. Additionally, the operating system provides checks of expected conditions in areas of code deemed sensitive. If an error is detected by this mechanism, a security function is initiated: the detected attack type is logged in a table; when the number of attacks reach a pre-set limit, the module initiates card termination, including overwriting of the CSPs, and the module is no longer operable.

- Side-channel attacks (SPA/DPA, timing analysis): the chip implements hardware countermeasures such that timing and current consumption are independent from processed data. The operating system enables the hardware countermeasures and implements independent countermeasures in code, such as constant time execution or randomized intermediate values.
- Card tearing attacks: the operating system implements methods to ensure protective measures are completed in the next cycle if the module loses power (i.e., if the power line is cut) before completion of the protective function.