FIPS Applet on RookySE

FIPS Applet v1.6.1.4 on RookySE ‘097153’

FIPS 140-3 Non-Proprietary Cryptographic Module Security Policy

Version: 1.2
Revision Date: 30/09/2024

About IDEMIA

OT-Morpho is now IDEMIA, the global leader in trusted identities for an increasingly digital world, with the ambition to empower citizens and consumers alike to interact, pay, connect, travel and vote in ways that are now possible in a connected environment.

Securing our identity has become mission critical in the world we live in today. By standing for Augmented Identity, we reinvent the way we think, produce, use and protect this asset, whether for individuals or for objects. We ensure privacy and trust as well as guarantee secure, authenticated and verifiable transactions for international clients from Financial, Telecom, Identity, Security and IoT sectors.

With close to €3bn in revenues, IDEMIA is the result of the merger between OT (Oberthur Technologies) and Safran Identity & Security (Morpho). This new company counts 14,000 employees of more than 80 nationalities and serves clients in 180 countries.

| For more information, visit www.idemia.com / Follow @IdemiaGroup on Twitter

Specifications and information are subject to change without notice.

The products described in this document are subject to continuous development and improvement.

All trademarks and service marks referred to herein, whether registered or not in specific countries, are the properties of their respective owners.

- Printed versions of this document are uncontrolled

TABLE OF CONTENT

TABLE OF ILLUSTRATIONS

TABLE OF TABLES

1 GENERAL

This document defines the Security Policy for FIPS Applet on RookySE with firmware FIPS Applet v1.6.1.4 on RookySE ‘097153’ cryptographic module. FIPS Applet on RookySE is a Hardware Security Module made by Idemia, hereafter denoted *the module*.

The module, validated to [NIST.FIPS.140-3] overall Level 3, meets security levels of following individual areas.

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 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

Table 1 Security Level

2 CRYPTOGRAPHIC MODULE SPECIFICATION

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 a secure element hardware with a tamper-protection.

2.1 Module Specifications

2.1.1 Module Type and 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. The module boundary is shown in red in the following picture:

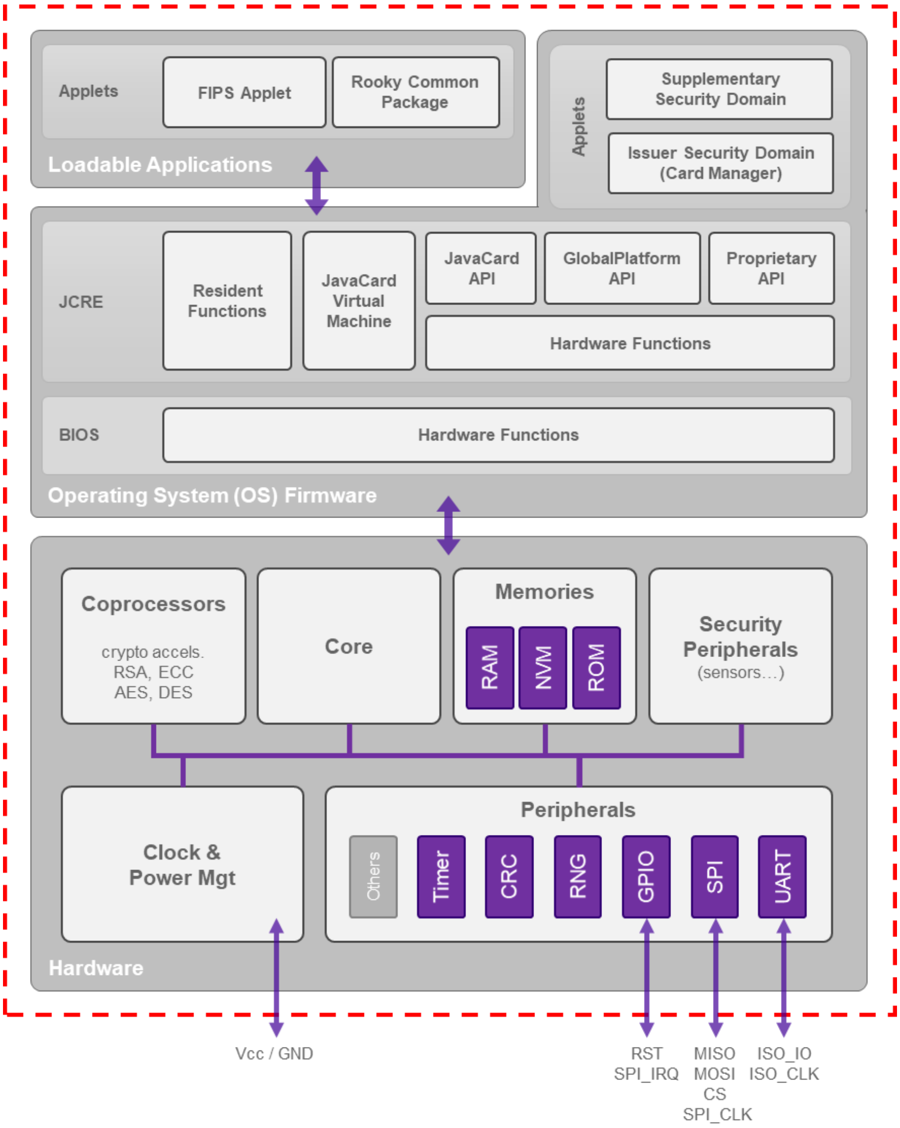

Figure 1 Module Boundary

The below picture 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):

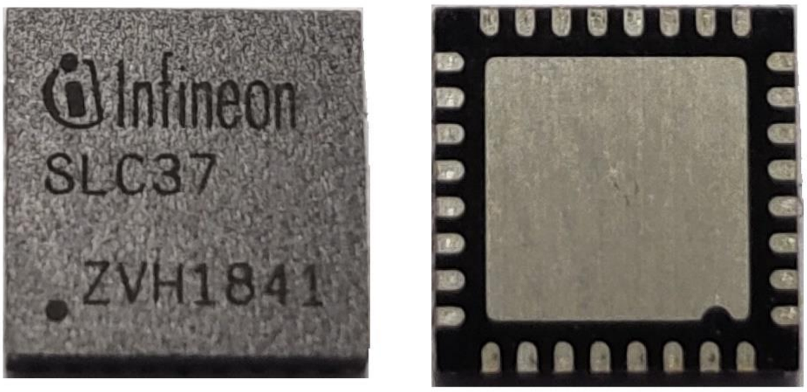

Figure 2 FIPS Applet on RookySE Chip

2.1.2 Module Components and Configuration

Below components are part of the module:

Model/Part Number	Hardware Version	Firmware Version	Processors
SLC37ESA2M0	Version: '29'	Global Platform OS: RookySE '097153' Javacard Application: FIPS Applet v1.6.1.4	32-bit ARM® SecurCore® SC300™

Table 2 Cryptographic Module Tested Configuration

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.

The module does not implement any Vendor Affirmed Operational Environments.

2.2 Modes of Operations

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 the section 4.3.2. A global indicator via FIPS Applet GET INFO service (unauthenticated service) is provided to show the status mode of operation.

2.2.1 Approved Mode of Operation

This mode means that the module is applying 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 4.3.2 are available for users.

In this mode, customer is still allowed to load additional firmware but this is restricted to firmware (applet) that is already validated under [NIST.FIPS.140-3]. Any firmware (applet) that is not validated [NIST.FIPS.140-3] will cancel the FIPS status of the module. A procedure for firmware loading is described in [FQR 401 9097 Ed 4] section 3.4.2.

2.3 Security Functions

The module implements the ‘approved security functions’ and ‘non-approved but allowed security functions’ listed in Table 3 and Table 4 respectively.

2.3.1 Approved Security Functions

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 3 are implemented by the module.

CAVP Cert	Algorithm and Standard	Mode/ Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2912	AES [NIST.FIPS.197] [NIST.SP.800-38A]	ECB, CBC	128/192/256	Data encryption/decryption
A2912	AES [NIST.SP.800-38B]	CMAC	128/192/256	MAC Generation/Verification, SP800-108 KDF
Vendor Affirmed	CKG [NIST.SP.800-133.Rev2]	5.1 Key Pairs for Digital Signature Schemes 5.2 Key Pairs for Key Establishment 6.1 The “Direct Generation” of Symmetric Keys 6.2.1 Symmetric Keys Generated Using Key-Agreement Schemes 6.2.2 Symmetric Keys Derived from a Pre-existing Key	RSA 2048, RSA 3072, RSA 4096, ECC P-224, ECC P-256, ECC P-384, ECC P-521, TDES-64, TDES-128, TDES-192, HMAC 64, HMAC 128, HMAC 160, HMAC 224, HMAC 256, HMAC 320, HMAC 384, HMAC 512, AES 128, AES 192, AES 256	Key generation Symmetric and Asymmetric
A2912	DRBG [NIST.SP.800-90A.Rev1]	CTR	256	Deterministic Random Bit Generation
A2912	ECDSA [NIST.FIPS.186-4]	CKG using method in section 4 and 5.1 [NIST.SP.800-133.Rev2]	P-224, P-256, P-384, P-521	ECDSA Key Generation,
A2912	ECDSA [NIST.FIPS.186-4]		P-192, P-224, P-256, P-384, P-521	ECDSA Key Verification
A2912	ECDSA [NIST.FIPS.186-4]	SHA2-224, SHA2-256, SHA2-384, SHA2-512	P-224, P-256, P-384, P-521	ECDSA Signature Generation
A2912	ECDSA [NIST.FIPS.186-4]	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	P-192, P-224, P-256, P-384, P-521	ECDSA Signature Verification

ENT (P)	Entropy Source [NIST.SP.800-90B]	Physical		Hardware True RNG used to seed the DRBG. The minimum number of bits of entropy is specified in the Table 17 Non-Deterministic Random Number Generation Specification
A2912	HMAC [NIST.FIPS.198-1]	SHA2-256 SHA2-384, SHA2-512	Key strength : 112 (minimum)	Message Authentication; SP800-108 KDF MAC Generation
A2912	HMAC [NIST.FIPS.198-1]	SHA-1, SHA2-256, SHA2-384, SHA2-512	Key strength : 64 (minimum)	Message Authentication; MAC Verification
A2912	KAS-ECC [NIST.SP.800-56A.Rev3]	with bilateral key confirmation	P-521 curve providing 256 bits of encryption strength	KAS-ECC SP800-56Ar3
A2912	KDF [NIST.SP.800-108]	AES CMAC, HMAC	HMAC-64, HMAC-128, HMAC-160, HMAC-224, HMAC-256, HMAC-320, HMAC-384, HMAC-512 AES-128, AES-192, AES-256	Key Derivation Symmetric Keys Derived from a Pre-existing Key using KDF
A2912	KTS (Secure Channel GP) [NIST.SP.800-38F]	SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	256 bits keys providing 256 bits of encryption strength	AES-CBC, AES-CMAC
A2912	KTS (Secure Session) [NIST.SP.800-38F]	SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	256 bits keys providing 256 bits of encryption strength	AES-CBC, AES-CMAC
A2912	KTS (Token transport) [NIST.SP.800-38F]	SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	128,192, 256 bits keys providing 128,192, 256 bits of encryption strength	AES-CBC, AES-CMAC
A2912	RSA [NIST.FIPS.186-4]	N/A	2048/3072/4096	RSA Key Generation
A2912	RSA [NIST.FIPS.186-4]	SHA2-224, SHA2-256, SHA2-384, SHA2-512	2048/3072/4096	RSA Signature Generation using PCKS1 v1.5 and PSS Scheme
A2912	RSA [NIST.FIPS.186-4]	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	1024/2048/3072/4096	RSA Signature Verification using PCKS1 v1.5 and PSS Scheme (SHA-1 and module 1024 allowed for legacy use)

A2912	RSA (CVL) [NIST.FIPS.186-4]		2048	RSA Signature Generation Primitive The RSA SigGen (CVL) shall only be used within the context of a FIPS 186-4 signature generation
A2912	SHA-3 [NIST.FIPS.202]	SHA3-224, SHA3-256, SHA3-384, SHA3-512	N/A	Message Digest
A2912	SHS [NIST.FIPS.180-4]	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	N/A	Message Digest

Table 3 Approved Algorithms

2.3.2 Non-approved but Allowed Security Functions

These algorithms do not claim any security and are not used to meet [NIST.FIPS.140-3] requirements. Therefore, SSPs do not map to these algorithms.

Algorithm	Caveat	Use / Function
CSPs obfuscation	(no security claimed)	CSPs obfuscation with a non-Approved algorithm

Table 4 Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed

The module does not implement any Non-Approved Algorithms Allowed in the Approved Mode of Operation with security claimed.

3 CRYPTOGRAPHIC MODULE INTERFACES

3.1 Physical and Logical Interface

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.

3.1.1 ISO7816 T0 Contact Protocol

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

Physical port	Logical interface	Data that passes over port/interface
VCC, GND	Power	ISO 7816: Supply voltage
RST	Control input	ISO 7816: Reset
CLK	Control input	ISO 7816: Clock
I/O	Control input, Status output Data input, Data output	ISO 7816: Input / Output of Data ISO 7816: Status Word ISO 7816: Procedure Byte

Table 5 Ports and Interfaces in ISO7816 T0 Contact Protocol

3.1.2 Serial Peripheral Interface (SPI) Protocol

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.

Physical port	Logical interface	Data that passes over port/interface
VCC, GND	Power	Supply voltage

RST	Control input	Chip Reset
SPI_CLK	Control input	Clock
SPI_MISO	Control input, Data input	Master IN, Slave OUT
SPI_MOSI	Data output, Status output	Master OUT, Slave IN
SPI_CS	Control input	Chip Select
SPI_IRQ	Status output	A status signal to inform command completion

Table 6 Ports and Interfaces in Serial Peripheral Interface (SPI) Protocol

4 ROLES, SERVICES, AND AUTHENTICATION

4.1 Roles

The FIPS Applet on RookySE Cryptographic module supports following user roles:

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*
Administrator	**CO**	***Crypto Officer Role 2*** *- Performing module initialization with the help SuperUser for its creation, global configuration, user management, and profile management for user and key*
Auditor	**AU**	***Crypto Officer Role 3*** *- Performing Audit Management*
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 a standard user A that have role to do Crypto functions but no role for key management*

Role Name	ID	Role Description
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

Table 7 Role Description

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.

4.2 Authentication

There are three types of authentication methods employed by the Module:

1. Global Platform Secure Channel Protocol '03' (GP SCP '03') Authentication
2. FIPS Applet Password-based Authentication
3. FIPS Applet Smartcard-based Authentication

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 logoff, 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.

4.2.1 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.

4.2.2 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 user name 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 success, 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 in order 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 request 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.

4.2.3 Authentication Strength in Each Role

Role	Authentication Method	Authentication Strength
Application Administrator	Identity-based authentication using Global Platform Secure Channel Protocol '03' Authentication	• **Single Attempt Probability** The probability that a random attempt will succeed using this authentication method is: • 1/(2^128) = 2.9E-39 (MAC||cryptogram, using a 128-bit block for authentication) **Multiple Attempts Probability** The module enforces a "slowdown mechanism" that increases the response time between two authentications attempts following a failed authentication, such that no more than nine (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) = 2.6E-38 (MAC||cryptogram, using a 128-bit block for authentication)

Role	Authentication Method	Authentication Strength
Administrator **Key Custodians** **Super User** **Auditor** **Standard User**	Identity-based authentication using Smartcard	**Single Attempt Probability** The probability that a random attempt will succeed using this authentication method is: • 1/(2^128) = 2.9E-39 (128-bit long cryptogram computed with 256-bit long key) **Multiple Attempts Probability** The module enforces a "slowdown mechanism" that increases the response time between two authentications attempts following a failed authentication, such that no more than nine (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) = 2.6E-38 (128-bit long cryptogram computed with 256-bit long key)
Administrator **Key Custodians** **Super User** **Auditor** **Standard User**	Identity-based authentication using Password	**Single Attempt Probability** The probability that a random attempt will succeed using this authentication method is: • 1/(2^128) = 2.9E-39 (128-bit long cryptogram computed with 256-bit long key) **Multiple Attempts Probability** The module enforces a "slowdown mechanism" that increases the response time between two authentications attempts following a failed authentication, such that no more than nine (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) = 2.6E-38 (128-bit long cryptogram computed with 256-bit long key)

Table 8 Role and Authentication

4.3 Services

All services implemented by the module are listed in the section **4.3.1** for each service description and its access for each Role. The module only provides approved services listed in the section 4.3.2 Approved

Services. 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.

4.3.1 Roles, Service Commands, Input and Ouput

The following table shows data input and output of each service employed by the module. 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 Table 7

Role	Service	Input	Output
Global Platform Services			
AA	GP Secure Channel	Key Identifier, Host Challenge, Security Level Host Cryptogram, MAC	Card Challenge, Card Cryptogram, diversification data, and key information, Status Word of the command
AA	GP Manage Content	Loaded package and stored data and key encrypted through GP SCP '03' Security Level 3	Status Word of the command
AA	GP Get Status	Application Type	List of application and its status, Status Word of the command
AA	GP Get Data	Data Tag	Corresponding data based on input tag
AA	GP Life Cycle	Status type and State Control	Status Word of the command
UU	GP Select	Application (applet) AID	Status Word of the command
UU	SE Reset	Signal Reset in port RST	Answer-to-RESET (ATR)
Firmware Upgrades Services - FIPS Applet			
AA	Initialize Update	Key Identifier and host challenge	Card Challenge, Card Cryptogram, diversification data, and key information, Status Word of the command
AA	External Authenticate	Security Level 3, Host Cryptogram and MAC	Status Word of the command
AA	Store Data	Sensitive Data and Key encrypted through GP SCP '03' Security Level 3	Status Word of the command
FIPS Applet Services			
UU	Get Info	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 active user, and/or last executed frame id Return Status[1]
UU	Manage Session	Session type	Return Status

[1] *Completion Status that indicates execution status result (success or error with specific reason code)*

Role	Service	Input	Output
UU	Mutual Authenticate	username, user/host public ephemeral key, selected authentication method, and host ID	System's public ephemeral key and receipt for key confirmation Return Status
UU	User Authenticate	user token generated from MAC of user's credential using generated session key from Key Agreement process in Mutual Authenticate	Authentication result Return Status
SU, CO, KC1, KC2, KC3, AU, UR	User Logout	Username information	Return Status
SU, CO, KC1, KC2, KC3, AU, UR	Secure Channel	Encrypted data and its signature	Encrypted response data Return Status
AU	Check Log	Number of logs to be returned	Activity logs and Return Status
AU	Read Log	Number of logs to be read and deleted	Activity logs and Return Status
AU	Delete Log	Number of logs to be deleted	Return Status
AU	Export Log (mode 01)	Mode to export the logs (mode: deleted)	All activity logs and Return Status
CO, AU	Export Log (mode 02)	Mode to export the logs (mode: kept)	All activity logs and Return Status
CO	Profile Create	Profile name and the access control	Return Status
CO	Profile Delete	Profile name and deletion option	Return Status
CO	Profile View	Only input frame with no additional input data	information of all profiles Return Status
CO	User Create	User information and credential data	Return Status
SU	User Create ADMIN	Admin's information and credential data	Return Status
CO	User Delete	Username information	Return Status
SU, CO	User Change Credential	Username and new user’s password or new user’s public key	Return Status
CO	User View (full)	User Type (All Users)	Information of all users Return Status
SU, CO	User View (Admin only)	User Type (Admin only)	information about admin user Return Status
CO, KC1	CCMK Import 1	Secret data and its key check value to import CCMK using split knowledge procedure	Return Status

Role	Service	Input	Output
CO, KC2	CCMK Import 2	Secret data and its key check value to import CCMK using split knowledge procedure	Return Status
CO, KC3	CCMK Import 3	Secret data and its key check value to import CCMK using split knowledge procedure	Key check value of CCMK Return Status
CO, KC1	CCMK Export 1	Only input frame with no additional input data	Secret data and its key check value of CCMK using split knowledge procedure Return Status
CO, KC2	CCMK Export 2	Only input frame with no additional input data	Secret data and its key check value of CCMK using split knowledge procedure Return Status
CO, KC3	CCMK Export 3	Only input frame with no additional input data	Secret data and its key check value of CCMK using split knowledge procedure Return Status
CO	System Set Config	00 to keep FIPS Certified Product 01 to change to Non FIPS Certified Product	Return Status
CO, UR	Import Key	Imported key in key token form protected by protection key whose label specified in key token's properties	Imported key in key token form protected by CCMK of the module as protection key Return Status
CO, UR	Export Key	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	Exported key in key token form with protection key as configured in input frame Return Status
CO, UR	Load Key	Loaded key in key token form protected by protection key whose label specified in key token's properties	Return Status
CO, UR	Diversify Key	Key token properties and diversification method	Diversified key in key token form protected by CCMK of the module as its protection key Return Status
CO, UR	Generate Key	Key token properties	Generated key in key token form protected by CCMK of the module as its protection key Return Status
CO, UR	Generate Key Pair	Key token properties	Generated key in key token form protected by CCMK of the module as protection key

Role	Service	Input	Output
			Return Status
CO	Generate Key (For Split Knowledge)	Customer key properties	Return Status
CO, UR	CK Export 1	Only input frame with no additional input data	Customer key's secret data, its KCV and properties Return Status
CO, UR	CK Export 2	Only input frame with no additional input data	Customer key's secret data and its KCV Return Status
CO, UR	CK Export 3	Only input frame with no additional input data	Exported Customer key's properties, secret data, secret KCV, customer key KCV Return Status
CO, UR	CK Import 1	Customer key's properties, secret data, and its KCV to import Customer Key using split knowledge procedure	Return Status
CO, UR	CK Import 2	Customer key's secret data and its KCV to import Customer Key using split knowledge procedure	Return Status
CO, UR	CK Import 3	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
CO	Set Date	Date information	Return Status
SU, CO	System Reset	Only input frame with no additional input data	Return Status
SU, CO	Get Data	Requested data such as: System, Host or Super User's Ephemeral public key	Requested data Return Status
SU, CO	System Store Data	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
SU, CO	Sync Request Token	Sync properties	Sync token data Return Status
SU, CO	Sync Write Token	Sync token data	Sync result status
SU, CO	Sync Report	Sync result data	Return Status
CO, UR	Crypto Encipher	Encipher Key label, plain data and cipher configuration	Encrypted data Return Status
CO, UR	Crypto Decipher	Decipher Key label, encrypted data and decipher configuration	Decrypted data Return Status

Role	Service	Input	Output
CO, UR	Crypto Sign	Signature Key label, data and signature configuration	Signature of the data Return Status
CO, UR	Crypto Verify	Signature Key label, data, signature of the data, and verification configuration	Signature verification result Return Status
CO	User Unblock	Username information	Return Status
SU	Admin Unblock	Admin username information	Return Status
SU, CO	System Unblock	Only input frame with no additional input data	Return Status

Table 9 Roles, Service Commands, Input and Output

4.3.2 Approved Services

Services listed below are approved services: either FIPS-approved security services, [ISO/IEC 19790] 7.4.3 services, non security-relevant services, or services using non-approved algorithm but claiming no security. Those services are available when the module is running in approved mode of operation.

Service	Description	Approved Security Function	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
GP Secure Channel	Establish a Global Platform secure communications channel	GP SCP ‘03’ Authentication and Secure Messaging vendor affirmed CKG [NIST.SP.800-133.Rev2] A2912	**OS-DRBG-STATE** **SD-KENC** **SD-KMAC** **SD-SENC** **SD-SMAC** **SD-RMAC**	**AA**	**G, E** **E** **E** **G, E** **G, E** **G, E**	**Completion Status**[2]
GP Manage Content	Load and install application packages and its associated keys and data	GP SCP ‘03’ Secure Messaging AES Encryption/Decryption A2912	**SD-KENC** **SD-KMAC** **SD-KDEK** **SD-SENC** **SD-SMAC** **SD-RMAC** **DAP-AES**	**AA**	**W** **W** **W** **E** **E** **E** **W, E**	**Completion Status**
GP Life Cycle	Set GP life cycle	-	**OS-MKEK**	**AA**	**Z**	**Completion Status**
SE Reset	Reset Warm/Cold	-	**H-eAUTH_sk** **H-Zs** **H-Ze** **H-SKAuthEnc** **H-SKAuthMac** **H-SKAuthKC** **H-SKSyncEnc** **H-SKSyncMac** **H-eAUTH_pk** **H-eHOST_pk** **H-HostID**	**UU**	**Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z**	**Completion Status**
Initialize Update	Performs initiation of a GP SCP ‘03’ Secure Channel Session	GP SCP ‘03’ Authentication vendor affirmed CKG [NIST.SP.800-133.Rev2] A2912	**SD-SENC** **SD-SMAC** **SD-RMAC**	**AA**	**G, E** **G, E** **G**	**Completion Status**
External Authenticate	Authenticate the host and to determine the level of security required for all subsequent commands	GP SCP ‘03’ Authentication A2912	**SD-SENC** **SD-SMAC** **SD-RMAC**	**AA**	**E** **E** **E**	**Completion Status**

[2] *Completion Status that indicates execution status result (success or error with specific reason code)*

Store Data	Transfer data to an Application in the GP secure channel	GP SCP ‘03’ Secure Messaging AES Encryption/Decryption A2912	**H-sAUTH_sk** **H-Ksync** **H-sAUTH_pk** **H-sHOST_pk** **H-sUSER_pk**	**AA**	**W** **W** **W** **W** **W**	**Completion Status**
Manage Session	FIPS Applet open session and close session	N/A	**H-KT_ECDSA_PAIR** **H-KT_ECDSA_sk** **H-KT_AES** **H-KT_3DES** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_HMAC** **H-KT_ECDSA_pk H-KT_RSA_pk**	**UU**	**Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z**	**Completion Status**
Mutual Authenticate	Public key exchange leads to key agreement between HOST’s operator and the module	C(2s,2e, ECDH) Key Agreement using Curve P521 One Step Key Derivation Counter Mode using HMAC-SHA2-256 A2912 vendor affirmed CKG [NIST.SP.800-133.Rev2]	**H-sAUTH_sk** **H-eAUTH_sk** **H-eD** **H-Zs** **H-Ze** **H-SKAuthEnc** **H-SKAuthMac** **H-SKAuthKC** **H-sAUTH_pk** **H-sHOST_pk** **H-sUSER_pk** **H-eAUTH_pk** **H-eHOST_pk** **H-HsmID** **H-HostID**	**UU**	**E** **G, E, Z** **E** **G, E, Z** **G, E, Z** **G** **G** **G, E** **E** **E** **E** **G, E, Z** **G, E, Z** **E** **W, E**	**Completion Status**
User Authenticate	Login on current open session	Key Confirmation using AES-CMAC-256 A2912	**H-SKAuthEnc** **H-SKAuthMac** **H-SKAuthKC** **H-User_pwd**	**UU**	**Z** **Z** **E, Z** **E**	**Completion Status**
Secure Channel	Secure Messaging for sensitive data	AES-256-CBC for Message Encryption & Decryption AES-256-CMAC for Message Authenticity A2912	**H-SKAuthEnc** **H-SKAuthMac** **H-SKAuthKC**	**SU, CO, KC1, KC2, KC3, AU, UR**	**E, Z** **E, Z** **E, Z**	**Completion Status**

CCMK Import 1	Import secret based on user's (KeyCustodian1) input	Split knowledge procedure and 32 Bits for KCV A2912	**N/A**	**CO, KC1**	**N/A**	**Completion Status**
CCMK Import 2	Import secret based on user's (KeyCustodian2) input	Split knowledge procedure and 32 Bits for KCV A2912	**N/A**	**CO, KC2**	**N/A**	**Completion Status**
CCMK Import 3	Import secret based on user's (KeyCustodian3) input. CCMK will be set in this sequence	Split knowledge procedure and 32 Bits for KCV A2912 One Step Key Derivation Counter Mode: AES-CMAC-256 vendor affirmed CKG [NIST.SP.800-133.Rev2] A2912	**H-CCMK** **H-CCMKEnc** **H-CCMKMac**	**CO, KC3**	**W, Z** **G, Z** **G, Z**	**Completion Status**
CCMK Export 1	Export secret of KeyCustodian1 Secret is calculated by FIPS applet in this case	vendor affirmed CKG (SP800-133 Rev2) Split knowledge procedure and 32 Bits for KCV A2912	**H-CCMK**	**CO, KC1**	**G**	**Completion Status**
CCMK Export 2	Export secret of KeyCustodian2. Secret is calculated by FIPS applet in this case	Split knowledge procedure and 32 Bits for KCV A2912	**H-CCMK**	**CO, KC2**	**E**	**Completion Status**
CCMK Export 3	Export secret of KeyCustodian3. Secret is calculated by FIPS applet in this case. CCMK will be set in this sequence	Split knowledge procedure and 32 Bits for KCV One Step Key Derivation Counter Mode: AES-CMAC-256 vendor affirmed CKG [NIST.SP.800-133.Rev2] A2912	**H-CCMK** **H-CCMKEnc** **H-CCMKMac**	**CO, KC3**	**E** **G** **G**	**Completion Status**

Import Key	Create new key with parameters totally based on user's input.	Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_pk** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_pk** **H-KT_ECDSA_sk**	**CO, UR**	**E** **E** **W, E** **W** **W** **W** **W** **W** **W** **W** **W** **W** **W**	**Completion Status**
Export Key	Export existing key into key file with CCMK or other existing key protection	Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_pk** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_pk** **H-KT_ECDSA_sk**	**CO, UR**	**E** **E** **R, E** **R** **R** **R** **R** **R** **R** **R** **R** **R** **R**	**Completion Status**
Load Key	Load PERMANENT key from LKD or VOLATILE FOR STORAGE key from LKD_DP into FIPS applet	Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_pk** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_pk** **H-KT_ECDSA_sk**	**CO, UR**	**E** **E** **W** **W** **W** **W** **W** **W** **W** **W** **W** **W** **W**	**Completion Status**

Diversify Key	Diversify an existing key with user's choice of method	Using approved Key Derivation Function SP800-108 Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1	**H-CCMKEnc** **H-CCMKMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC**	**CO, UR**	**E** **E** **G, E** **G** **G, E**	**Completion Status**
Generate Key	Create a new key with user's input algorithm Key value randomized by FIPS applet	vendor affirmed CKG [NIST.SP.800-133.Rev2] Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC**	**CO, UR**	**E** **E** **G** **G** **G**	**Completion Status**
Generate Key Pair	Create a new asymmetric key User can freely choose the algorithm and curve Key value randomized by FIPS applet	vendor affirmed CKG [NIST.SP.800-133.Rev2] Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_ECDSA_PAIR**	**CO, UR**	**E** **E** **G** **G** **G**	**Completion Status**
Generate Key (for Split Knowledge)	Customer key generation This service will determine the specification of key token output from the whole Customer Key Ceremony procedure	vendor affirmed CKG [NIST.SP.800-133.Rev2]	**H-KT_AES** **H-KT_3DES** **H-KT_HMAC**	**CO**	**G** **G** **G**	**Completion Status**
CK Export 1	Export a secret for customer key creation for KCP1	Split knowledge procedure and 32 bits KCV A2912	**H-KT_AES** **H-KT_3DES** **H-KT_HMAC**	**CO, UR**	**E** **E** **E**	**Completion Status**

CK Export 2	Export a secret for customer key creation for KCP2	Split knowledge procedure and 32 bits KCV A2912	**H-KT_AES** **H-KT_3DES** **H-KT_HMAC**	**CO, UR**	**E** **E** **E**	**Completion Status**
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	Split knowledge procedure and 32 bits KCV AES-CMAC-256 for Key Authenticity A2912	**H-KT_AES** **H-KT_3DES** **H-KT_HMAC** **H-CCMKMac**	**CO, UR**	**E** **E** **E** **E**	**Completion Status**
CK Import 1	Import secret based on user's (KCP1) input	Split knowledge procedure and 32 bits KCV	**N/A**	**CO, UR**	**N/A**	**Completion Status**
CK Import 2	Import secret based on user's (KCP2) input	Split knowledge procedure and 32 bits KCV	**N/A**	**CO, UR**	**N/A**	**Completion Status**
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	Split knowledge procedure and 32 bits KCV Key Protection using combination of approved encryption method and approved authentication method in section 2.3.1 A2912	**H-CCMKEnc** **H-CCMKMac**	**CO, UR**	**E** **E**	**Completion Status**
FIPS Applet Store Data	This service normally used to store SUPER_USER public key within FIPS Applet Initialization procedure after SUPER_USER change its smartcard's PIN	ECC CDH Key Agreement for Key Validation Message Digest SHA2-256 A2912	**H-sHOST_pk** **H-sUSER_pk**	**SU, CO**	**W** **W**	**Completion Status**

Sync Request Token	This service is targeted to SE_MASTER to retrieve a sync token which will be referred on Sync Write Token service	Key Derivation One Step Counter Mode : AES-CMAC-256 vendor affirmed CKG [NIST.SP.800-133.Rev2] **Automated SSPs Establishment using approved method SP800-38F** SYNC Message Encryption: AES-256-CBC-M2 SYNC Message Authenticity: AES-256-CMAC A2912	**H-KSync** **H-SKSyncEnc** **H-SKSyncMac**	**SU, CO**	**E** **G, E, Z** **G, E, Z**	**Completion Status**
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	Key Derivation One Step Counter Mode: AES-CMAC-256-M2 vendor affirmed CKG [NIST.SP.800-133.Rev2] **Automated SSPs Establishment using approved method SP800-38F** SYNC Message Decryption & Encryption : AES-256-CBC SYNC Message Authenticity: AES-256-CMAC A2912	**H-KSync** **H-SKSyncEnc** **H-SKSyncMac** **H-KT_AES** **H-KT_3DES** **H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_pk** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_pk** **H-KT_ECDSA_sk**	**SU, CO**	**E** **G, E, Z** **G, E, Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z**	**Completion Status**
Sync Report	This service purpose is to confirm that Sync Write Token had performed successfully. The target of this service is SE_MASTER	SYNC Message Decryption: AES-256-CBC-M2 SYNC Message Authenticity: AES-256-CMAC A2912	**H-SKSyncEnc** **H-SKSyncMac**	**SU, CO**	**E, Z** **E, Z**	**Completion Status**

Crypto Encipher	Perform encipher on the user's input data	When used with an approved encryption algorithm in section 2.3.1 A2912	**H-KT_AES**	**CO, UR**	**E**	**Completion Status**
Crypto Decipher	Perform decipher on the user's input data	When used with an approved decryption algorithm in section 2.3.1 A2912	**H-KT_AES**	**CO, UR**	**E**	**Completion Status**
Crypto Sign	Create signature based on the user's input data	When used with an approved signature and MAC algorithm in section 2.3.1 A2912	**H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_sk**	**CO, UR**	**E** **E** **E** **E** **E** **E** **E**	**Completion Status**
Crypto Verify	Confirm the verified status of the user's signature data	When used with an approved verification and MAC algorithm in section 2.3.1 A2912	**H-KT_HMAC** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_pk** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_pk**	**CO, UR**	**E** **E** **E** **E** **E** **E**	**Completion Status**
FIPS Applet Set Config	Set FIPS applet configuration	Service that uses processes in approved manner	**N/A**	**CO**	**N/A**	**Completion Status**
Profile Create	Create new profile	Service that uses processes in approved manner	**N/A**	**CO**	**N/A**	**Completion Status**
Profile Delete	Delete non-default profile based on profile name input	Service that uses processes in approved manner	**N/A**	**CO**	**N/A**	**Completion Status**
User Create	Create new user	Service that uses processes in approved manner	**H-User_pwd** **H-sUSER_pk**	**CO**	**W** **W**	**Completion Status**
User Create ADMIN	Admin creation. This is a step during FIPS Applet Initialization procedure	Service that uses processes in approved manner	**H-User_pwd** **H-sUSER_pk**	**SU**	**W** **W**	**Completion Status**

User Delete	Delete user based on user name input	Service that uses processes in approved manner	**H-User_pwd** **H-sUSER_pk**	**CO**	**Z** **Z**	**Completion Status**
User Change Credential	Change user's password or user's public key	Service that uses processes in approved manner	**H-User_pwd** **H-sUSER_pk**	**SU, CO**	**W** **W**	**Completion Status**
User Logout	Logout the user	**N/A**	**H-SKAuthEnc** **H-SKAuthMac** **H-SKAuthKC**	SU, CO, KC1, KC2, KC3, AU, UR	**Z** **Z** **Z**	**Completion Status**
System Reset	Reset admin user, applet life cycle, LKD, and LKD_DP	N/A	**H-CCMK** **H-CCMKEnc** **H-CCMKMac** **H-User_pwd** **H-KT_ECDSA_PAIR** **H-KT_ECDSA_sk** **H-KT_AES** **H-KT_3DES** **H-KT_RSA_CRT_PAIR** **H-KT_RSA_SFM_PAIR** **H-KT_RSA_CRT_sk** **H-KT_RSA_SFM_sk** **H-KT_HMAC** **H-KT_ECDSA_pk** **H-KT_RSA_pk** **H-sUSER_pk**	**SU, CO**	**Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z** **Z**	**Completion Status**

Table 10 Approved Security Services

G = Generate: The module generates or derives the SSP.
R = Read: The SSP is read from the module (e.g. the SSP is output).
W = Write: The SSP is updated, imported, or written to the module.
E = Execute: The module uses the SSP in performing a cryptographic operation.
Z = Zeroise: The module zeroises the SSP.
- = Not accessed by the service

Service	Approved Service’s Rationale
Global Platform Services	
GP Get Status	Show status service as specified in the [ISO/IEC 19790] section 7.4.3
GP Get Data	Show module’s versioning information service as specified in the ISO19790:2012 section 7.4.3
GP Select	There is no intervention to the security process nor access to defined SSP
FIPS Applet Services	
Get Info	Show module’s versioning information and Show status service as specified in the [ISO/IEC 19790] section 7.4.3
Check Log	There is no intervention to the security process nor access to defined SSP
Read Log	There is no intervention to the security process nor access to defined SSP
Delete Log	There is no intervention to the security process nor access to defined SSP
Export Log (mode 01)	There is no intervention to the security process nor access to defined SSP
Export Log (mode 02)	There is no intervention to the security process nor access to defined SS.
Profile View	There is no intervention to the security process nor access to defined SSP
User View (full)	There is no intervention to the security process nor access to defined SSP
User View (Admin only)	There is no intervention to the security process nor access to defined SSP
Set Date	There is no intervention to the security process nor access to defined SSP
Get Data	There is no intervention to the security process nor access to defined SSP
User Unblock	Crypto Officer management function service
Admin Unblock	Crypto Officer management function service
System Unblock	Crypto Officer management function service

Table 11 Approved Services - ISO19790:2012 7.4.3 Services, Non Security-Relevant Services, or Services Using Non-Approved Algorithm but Claiming No security

The module does not support any non-approved services.

4.3.3 Firmware Loading

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 10.2.2 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 the [NIST.FIPS.140-3] CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires a separate [NIST.FIPS.140-3] validation. A procedure for firmware loading is described in [FQR 401 9097 Ed 4] section 3.4.2.

5 SOFTWARE AND FIRMWARE SECURITY

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.

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 10.1.1 integrity self test.

5.1 Form of Executable Code

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
	Rooky Common Package	A binary code written in Java running on JCVM

Table 12 Form of Executable Code

5.2 Initiate on Demand

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

6 OPERATIONAL ENVIRONMENT

Not Applicable

(***Remarks.*** *The module is designated as a limited operational environment under the* [NIST.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 the* [NIST.FIPS.140-3] *CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires a separate* [NIST.FIPS.140-3] *validation.)*

7 PHYSICAL SECURITY

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:

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
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 13 Physical Security Inspection Guidelines

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

	Temperature or voltage measurement	Specify EFP[11] or EFT	Specify if this condition results in a shutdown or zeroisation
Low Temperature	-25°C	EFT	Shutdown
High Temperature	+85°C	EFT	Shutdown
Low Voltage	1.40V < Vcc < 1.62V	EFT	Shutdown
High Voltage	5.5V < Vcc < 7.9V	EFT	Shutdown

Table 14 EFP/EFT

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

	Hardness tested temperature measurement
Low Temperature	-25°C
High Temperature	+85°C

Table 15 Hardness Testing Temperature Range

8 NON-INVASIVE SECURITY

Not Applicable

9 SENSITIVE SECURITY PARAMETER MANAGEMENT

9.1 SSP Management

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
CRITICAL SECURITY PARAMETERS – OS								
OS-DRBG-SEED *Entropy input and nonce provided by the ENT (P)*	N/A	ENT (P)	Generated by a call to ENT when the DRBG is instantiated or when a reseed is required	N/A	N/A	Plaintext/ dynamic in RAM (Stack)	Overwrite with all zeros value on power cycle or when no longer used	Used to seed the Approved DRBG
OS-DRBG-STATE *The current AES-256 CTR_DRBG state*	N/A	N/A	Generated every time DRBG is generated	N/A	N/A	Plaintext/dynamic in RAM (Global)	Overwrite with all zeros value on power cycle or when no longer used	Store the state of CTR_DRBG
SD-KENC *Master Encryption Security Domain key* AES Key Mode: N/A (only derivation)	AES-256	AES A2912	N/A	Imported using APDU Command STORE DATA or PUT KEY through GP SCP '03' at Manufacturing, and also later, IN USE phase I/O type: electronic	N/A	Plaintext(obfuscated)/static in persistent memory of the Module at manufacturing stage	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM APDU Command GP Delete Key	Master key used to generate SD-SENC

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
SD-KMAC *Master MAC Security Domain key* AES Key Mode: N/A (only derivation)	AES-256	AES A2912	N/A	Imported using APDU Command STORE DATA or PUT KEY through GP SCP ‘03’ at Manufacturing I/O type: electronic	N/A	Plaintext(obfuscated)/static in persistent memory of the Module at manufacturing stage	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM APDU Command GP Delete Key	Master key used to generate SD-SMAC
SD-KDEK *Master DEK Security Domain key* AES Key Mode: N/A (only derivation)	AES-256	AES A2912	N/A	Imported using APDU Command STORE DATA or PUT KEY through GP SCP ‘03’ at Manufacturing I/O type: electronic	N/A	Plaintext(obfuscated)/static in persistent memory of the Module at manufacturing stage	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM APDU Command GP Delete Key	Sensitive data decryption key used to decrypt CSPs (SD-KENC, SD-KMAC, SD-KDEK, DAP-AES)
SD-SENC *GP Secure Channel Session Encryption Key* AES Key Mode: CBC	AES-256	CKG AES A2912	N/A	N/A	Generated during Secure Channel opening (in InitUpdate command) using approved key generation function (CKG)	Plaintext/dynamic as a volatile Native Key Object (RAM) inside the Module	Overwrite with all zeros value on: Power_ON or applet selection	Session encryption key used to encrypt / decrypt secure channel data

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
SD-SMAC *GP Secure Channel Session Command MAC Key* AES Key, Mode: CMAC	AES-256	CKG AES A2912	N/A	N/A	Generated during Secure Channel opening (in InitUpdate command) using approved key generation function (CKG)	Plaintext/dynamic as a volatile Native Key Object (RAM) inside the Module	Overwrite with all zeros value on: Power_ON or applet selection	Session MAC key used to verify inbound secure channel data integrity
SD-RMAC *GP Secure Channel Session Response MAC Key* AES Key Mode: CMAC	AES-256	CKG AES A2912	N/A	N/A	Generated during Secure Channel opening (in InitUpdate command) using approved key generation function (CKG)	Plaintext/dynamic as a volatile Native Key Object (RAM) inside the Module	Overwrite with all zeros value on: Power_ON or applet selection	Session MAC key used to generate response secure channel data MAC
DAP-AES *Data Authentication Pattern AES Key* AES Key Mode: CMAC	AES-128	AES A2912	N/A	Imported using APDU Command PUT KEY through GP SCP '03' at Manufacturing I/O type: electronic	N/A	Plaintext/static as persistent JCVM Key Object owned by the ISD inside the Module	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM APDU Command GP Delete Key	Used to calculate signature (MAC) of loaded package for firmware loading
CRITICAL SECURITY PARAMETERS – FIPS APPLET on RookySE								
H-sAUTH_sk	Curve P-521	KAS-ECC	N/A	Imported using APDU Command 'Store	N/A	Plaintext(obfuscated)/static as	Overwrite with all zeros value	Used in the Key Agreement together

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
System Authentication Static Private Key EC Private Key		A2912		Data' that is authenticated and protected using GP SCP '03' level 3 (Encrypted and MAC) on: Personalization at Manufacturing Firmware Loading process to upgrade version of new validated FIPS Applet I/O type: electronic		JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	on: Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	with *HOST Authentication Static Public Key* (**H-sHOST_pk**) or *User Authentication Static Public Key* (**H-sUSER_pk**) to generate shared secret **H-Z$_s$**
H-eAUTH_sk *System Authentication Ephemeral Private Key* EC Private Key	Curve P-521	CKG KAS-ECC A2912	Generated using ECDSA Key Pair Generation function (CKG) on Mutual Authenticate service	N/A	N/A	Plaintext/dynamic as JCVM Key Object owned by the FIPS Applet inside the Module as volatile data	Overwrite with all zeros value on: Destroy on shared secret **H-Z$_e$** generation. Any failure during mutual authentication On RESET (Warm/Cold)	Used in the Key Agreement together with *HOST Authentication Ephemeral Public Key* (**H-eHOST_pk**) to generate shared secret **H-Z$_e$**

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
H-Z$_s$ *Shared secret Zs* 66 bytes of secret data	N/A	KAS-ECC A2912	N/A	N/A	Established in the Key Agreement (A2912) between **H-sAUTH_sk** and **H-sUSER_pk** if authentication method is smartcard-based or **H-sHOST_pk** if authentication method is password-based on Mutual Authenticate service	Plaintext/Dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros value on: After Key Derivation Any failure during authentication On RESET (Warm/Cold)	Combined with **H-Z$_e$** to construct secret Z used in Key Derivation to generate session keys **H-SKAuthEnc**, **H-SKAuthMac, H-SKAuthKC** on successful user authentication
H-Z$_e$ *Shared secrets Ze* 66 bytes of secret data	N/A	KAS-ECC A2912	N/A	N/A	Established in the Key Agreement (A2912) between **H-eAUTH_sk** and **H-eHOST_pk** on Mutual Authenticate service	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros value on: After Key Derivation Any failure during authentication On RESET (Warm/Cold)	Combined with **H-Z$_s$** to construct secret Z used in Key Derivation to generate session keys **H-SKAuthEnc**, **H-SKAuthMac, H-SKAuthKC** on successful user authentication

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
H-SKAuthEnc *Encryption Secure Channel Session Key* AES Key Mode: CBC	AES-256	CKG AES A2912	N/A	N/A	Generated using approved key generation (CKG) using derivation from Key Agreement Scheme followed by One Step KDF in counter mode with (**H-Z$_s$** ||**H-Z$_e$**) as message in the process on Mutual Authenticate service	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros value on: Any failure during User authentication Error secure messaging in secure channel User logging out Close Session On RESET (Warm/Cold)	Used to encrypt and decrypt the message for secure messaging in the Secure Channel
H-SKAuthMac *MAC Secure Channel Session Key* AES Key Mode: CMAC	AES-256	CKG AES A2912	N/A	N/A	Generated using approved key generation (CKG) using derivation from Key Agreement Scheme followed by One Step KDF counter mode with (**H-Z$_s$** ||**H-Z$_e$**) as message in the process on Mutual	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros value on: Any failure during User authentication Error secure messaging in secure channel User logging out Close Session	Used to calculate MAC of the message for secure messaging in the Secure Channel

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
					Authenticate service		On RESET (Warm/Cold)	
H-SKAuthKC *Key Confirmation Secure Channel Session Key* AES Key Mode: CMAC	AES-256	CKG AES A2912	N/A	N/A	Generated using approved key generation (CKG) using derivation from Key Agreement Scheme followed by One Step KDF counter mode with (**H-Z_s** ||**H-Z_e**) as message in the process on Mutual Authenticate service	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros value on: Any failure during User authentication Error secure messaging in secure channel User logging out Close Session On RESET (Warm/Cold)	Used for bilateral key confirmation including for generating UserToken from username, {**H-User_pwd**}, **H-HsmID, H-HostID, H-eAUTH_pk** , and **H-eHOST_pk** in the authentication process Used to generate IVs for Message Encryption / Decryption for secure messaging in Secure Channel
H-CCMK *Crypto Card Master Key* AES Key Mode: N/A (only derivation)	AES-256	CKG AES DRBG KDF A2912	Generated internally using approved CKG (Direct Generation of Symmetric Key) and split into 3 secrets on Export CCMK services	Exported and encrypted through secure channel using split knowledge procedure on Export CCMK services Imported through secure channel using	Established from three different secrets from three different key custodians through secure channel on Import CCMK services	Plaintext(obfuscated)/static as JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	Overwrite with all zeros on: System Reset service Personalization at Manufacturing by erasing all	Used as master key to derrive **H-CCMKEnc** and **H-CCMKMac**

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
				split knowledge procedure on Import CCMK services I/O type: electronic			data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	
H-CCMKEnc *Encryption Derived CCMK Key* AES Key Mode : CBC	AES-256	CKG AES A2912	N/A	N/A	Derived from pre-existing key **H-CCMK** using one step KDF counter mode on successful completion of 3 secrets export or import	Plaintext(obfuscated)/static as JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	Overwrite with all zeros on: System Reset service Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	Used to encrypt/decrypt key token for Key Protection
H-CCMKMac *MAC Derived CCMK Key* AES Key Mode : CMAC	AES-256	CKG AES A2912	N/A	N/A	Derived from pre-existing key **H-CCMK** using one step KDF counter mode on successful completion of 3	Plaintext(obfuscated)/static as JCVM Key Object owned by the FIPS Applet inside the Module as	Overwrite with all zeros on: System Reset service Personalization at Manufacturing	Used to calculate MAC of key token as part of token for Key Protection

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
					secrets export or import	non-volatile data	by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	
H-KSync *Synchronization Master Key* AES Key Mode: N/A (only derivation)	AES-256	KDF A2912	N/A	Imported using APDU Command 'Store Data' that is authenticated and protected using GP SCP '03' level 3 (Encrypted and MAC) on: Personalization at Manufacturing Firmware Loading process to upgrade version of new validated FIPS Applet I/O type: electronic	N/A	Plaintext(obfuscated)/static in persistent byte array owned by FIPS Applet inside the module	Overwrite with all zeros on: Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	Used as master key to derive **H-SKSyncEnc** and **H-SKSyncMac** for Synchronization Session Keys
H-SKSyncEnc *Encryption Synchronization Session Key*	AES-256	CKG AES A2912	N/A	N/A	Derived from pre-existing key **H-KSync** using one step KDF counter mode with incremental	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros on: Any failure during	Used to encrypt/decrypt SYNC Token using Automated SSP Establishment Key Transport in

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
AES Key Mode: CBC					SYNC session counter as part of the message on every synchronization initiation (SYNC Request Token service in System MASTER and SYNC Write Token service in System SLAVE)		synchronization session After synchronization session completed On RESET (Warm/Cold)	synchronization process
H-SKSyncMac *MAC Synchronization Session Key* AES Key Mode: CMAC	AES-256	CKG AES A2912	N/A	N/A	Derived from pre-existing key **H-KSync** using one step KDF counter mode with incremental SYNC session counter as part of the message on every synchronization initiation (SYNC Request Token service in System MASTER and SYNC Write Token service in System SLAVE)	Plaintext/dynamic in transient byte array owned by FIPS Applet inside the module	Overwrite with all zeros on: Any failure during synchronization session After synchronization session completed On RESET (Warm/Cold)	Used to calculate MAC of SYNC Token using Automated SSP Establishment Key Transport in synchronization process
H-User_pwd	Maximum 16 char	AES	N/A	Imported along with username via User	N/A	Plaintext(obfuscated)/static in	Overwrite with all zeros on:	Used as part of credential data to

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
User's password credential Maximum 16 characters		KAS-ECC A2912		Create service through secure channel Updated the old password with new password via User Change Password service through secure channel I/O type: electronic		persistent byte array owned by FIPS Applet inside the module	System Reset User deletion Synchronization Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	generate User Token using **H-SKAuthKC**
H-KT_ECDSA_PAIR *Key Token ECDSA PAIR* ECC Key Pair	Curve P-192 Curve P-224 Curve P-256 Curve P-384 Curve P-521	CKG ECDSA A2912	Generated using ECDSA Key Pair Generation function (CKG) on Generate Key service	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset	**H-KT_ECDSA_PAIR** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) Public Key of this key token can be used to perform

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							- Manage session - Sync write token	Crypto operation via Crypto Verify service Private Key of this key token can be used to perform Crypto operation via Crypto Sign service **H-KT_ECDSA_PAIR** with Curve P-192 can only be used for signature verification and not for signature generation
H-KT_ECDSA_sk *Key Token ECDSA PRIVATE* ECC Private Key	Curve P-192 Curve P-224 Curve P-256 Curve P-384 Curve P-521	ECDSA A2912	N/A	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session	**H-KT_ECDSA_sk** can be stored outside the system module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) All curves (except curve P-192) of this key token can be used to perform Crypto operation via Crypto Sign service

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							- Sync write token	
H-KT_AES *Key Token AES* AES Key Mode: CBC	AES-128 AES-192 AES-256	CKG AES A2912	Generated internally using approved Generate Key service (CKG) – Direct Generation of Symmetric Key	Imported via Import Key service and Exported via Export Key service I/O type: electronic	**H-KT_AES** also can be generated from Diversify Key service using KDF Counter mode with approved MasterKey (**H-KT_AES** or **H-KT_HMAC**)	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session - Sync write token	**H-KT_AES** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) **H-KT_AES** can be used to protect another key token for external key storage (outside of System Module). This key token can be used to perform Crypto operation via Crypto Cipher, Crypto Decipher service.
H-KT_RSA_CRT_PAIR *Key Token RSA CRT PAIR* RSA CRT Key Pair Mode: PSS or PKSC1	RSA-1024 RSA-2048 RSA-3072 RSA-4096	CKG RSA A2912	Generated using RSA Key Pair Generation function (CKG) on approved Generate Key service	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and	**H-KT_RSA_CRT_PAIR** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session - Sync write token	approved symmetric key token (**H-KT_AES**) Public Key of this key token can be used to perform Crypto operation via Crypto Verify service. Private Key of this key token can be used to perform Crypto operation via Crypto Sign service. (Key Token RSA-1024 can only be used for signature verification)
H-KT_RSA_SFM_PAIR *Key Token RSA SFM PAIR* RSA SFM Key Pair Mode: PSS or PKSC1	RSA-1024 RSA-2048 RSA-3072 RSA-4096	CKG RSA A2912	Generated using RSA Key Pair Generation function (CKG) on approved Generate Key service	Imported via Import Key service and Exported via Export Key service. I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following	**H-KT_RSA_SFM_PAIR** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**)

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							services are executed: - System Reset - Manage session - Sync write token	Public Key of this key token can be used to perform Crypto operation via Crypto Verify service. Private Key of this key token can be used to perform Crypto operation via Crypto Sign service. (Key Token RSA-1024 can only be used for signature verification)
H-KT_RSA_CRT_sk *Key Token RSA CRT PRIVATE* RSA CRT Private Key Mode: PSS or PKSC1	RSA-1024 RSA-2048 RSA-3072 RSA-4096	RSA A2912	N/A	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset	**H-KT_RSA_CRT_sk** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) Key Token RSA Private can be used to perform Crypto

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							- Manage session - Sync write token	operation via Crypto sign service (For Crypto sign, key token RSA Private 1024 is excluded)
H-KT_RSA_SFM_sk *Key Token RSA SFM PRIVATE* RSA SFM Private Key Mode: PSS or PKSC1	RSA-1024 RSA-2048 RSA-3072 RSA-4096	RSA A2912	N/A	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session - Sync write token	**H-KT_RSA_SFM_sk** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) Key Token RSA Private can be used to perform Crypto operation via Crypto sign service. (For Crypto sign, key token RSA Private 1024 is excluded)
H-KT_HMAC	HMAC-64 HMAC-128 HMAC-160 HMAC-224	CKG HMAC	Generated internally using approved Generate Key	Imported via Import Key service and	**H-KT_HMAC** also can be generated from	Plaintext/dynamic as a volatile data	Key token data will be zeroised when the user is	**H-KT_HMAC** can be stored outside the System module

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
Key Token HMAC HMAC Key	HMAC-256 HMAC-320 HMAC-384 HMAC-512	A2912	service (CKG) – Direct Generation of Symmetric Key	Exported via Export Key service I/O type: electronic	Diversify Key service using KDF Counter mode with approved MasterKey (**H-KT_AES** or **H-KT_HMAC**)	inside the System module	blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session - Sync write token	protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**). This key token can be used to perform Crypto operation via Crypto Sign and Crypto Verify service; HMAC-64 is used only for verification (not HMAC generation)
PUBLIC SECURITY PARAMETER								
H-KT_ECDSA_pk *Key Token ECDSA PUBLIC* ECC Public Key	Curve P-192 Curve P-224 Curve P-256 Curve P-384 Curve P-521	ECDSA A2912	N/A	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset	**H-KT_ECDSA_pk** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) This key token can be used to perform Crypto operation Crypto Verify service

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							- Manage session Sync write token	
H-KT_RSA_pk *Key Token RSA PUBLIC* RSA Public Key Mode: PSS or PKSC1	RSA-1024 RSA-2048 RSA-3072 RSA-4096	RSA A2912	N/A	Imported via Import Key service and Exported via Export Key service I/O type: electronic	N/A	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed: - System Reset - Manage session - Sync write token	**H-KT_RSA_pk** can be stored outside the System module protected by **H-CCMKEnc** and **H-CCMKMac**, or by approved symmetric key token (**H-KT_AES**) Key Token RSA Public can be used to perform Crypto operation via Crypto Verify service. Key Token RSA-1024 Public can only be used for signature verification
H-sAUTH_pk *System Authentication Static Public Key* EC Public Key	Curve P-521	KAS-ECC A2912	N/A	Imported using APDU Command 'Store Data' that is authenticated and protected using GP SCP '03' level 3 (Encrypted and MAC) on:	N/A	Plaintext(obfuscated)/static as JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM	Used in the Key Agreement together with *HOST Authentication Static Private Key* or *User Authentication Static Private Key* to generate shared secret in **H-Z$_S$** in the client side

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
				Personalization at Manufacturing Firmware Loading process to upgrade version of new validated FIPS Applet Exported using System Get Data Service through Secure Channel I/O type: electronic			Firmware Loading Process using APDU Command Delete Package and Instance	
H-sHOST_pk *HOST Authentication Static Public Key* EC Public Key	Curve P-521	KAS-ECC A2912	N/A	Imported using APDU Command 'Store Data' that is authenticated and protected using GP SCP '03' level 3 (Encrypted and MAC) on: Personalization at Manufacturing Firmware Loading process to upgrade version of new validated FIPS Applet Exported using System Get Data	N/A	Plaintext/static as JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	Overwrite with all zeros value on: Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	Used in the Key Agreement together with *System Authentication Static Private Key* (**H-sAUTH_sk**) to generate shared secret **H-Z**$_{S}$

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
				Service through Secure Channel I/O type: electronic				
H-sUSER_pk *User Authentication Static Public Key* EC Public Key	Curve P-521	KAS-ECC A2912	N/A	Imported via User Create service through secure channel (for all users except Super User) Personalized at Manufacturing through GP SCP ‘03’ level 3 (only applicable for Super User’s Public Key) Updated using System Store Data service through secure channel (only applicable for Super User’s Public Key) Exported using System Get Data Service through Secure Channel (only applicable for Super User’s Public Key) I/O type: electronic	N/A	Plaintext(obfuscated)/static as JCVM Key Object owned by the FIPS Applet inside the Module as non-volatile data	Overwrite with all zeros on: System Reset User deletion Synchronization Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	Used in the Key Agreement together with *System Authentication Static Private Key* (**H-sAUTH_sk**) to generate shared secret **H-Z$_S$**

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
H-eAUTH_pk *System Authentication Ephemeral Public Key* EC Public Key	Curve P-521	CKG ECDSA [A2912](A2912)	Generated using ECDSA Key Pair Generation function (CKG) on Mutual Authenticate service	Returned to the user as part of response data of Mutual Authenticate service I/O type: electronic	N/A	Plaintext/dynamic as JCVM Key Object owned by the FIPS Applet inside the Module as volatile data	Overwrite with all zeros value on: Any failure during mutual authentication Successful user authentication On RESET (Warm/Cold)	Used in the Key Agreement together with *HOST Authentication Ephemeral Private Key* to generate shared secret in **H-Z_e** in the client side
H-eHOST_pk *HOST / SMA Authentication Ephemeral Public Key* EC Public Key	Curve P-521	KAS-ECC KDF [A2912](A2912)	N/A	Imported through Mutual Authenticate service I/O type: electronic	N/A	Plaintext/dynamic as JCVM Key Object owned by the FIPS Applet inside the Module as volatile data	Overwrite with all zeros value on: Any failure during mutual authentication Successful user authentication On RESET (Warm/Cold)	Used in the Key Agreement together with *System Authentication Ephemeral Private Key* (**H-eAUTH_sk**) to generate shared secret **H-Z_e**
H-HsmID *Predefined System Identification data*	N/A	KDF [A2912](A2912)	N/A	Pre-defined value	N/A	Plaintext/static in persistent byte array owned by the	Overwrite with all zeros value on:	Used as FixedInfo data in Key Confirmation and UserToken

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
						FIPS Applet inside the Module	Personalization at Manufacturing by erasing all data in NVM Firmware Loading Process using APDU Command Delete Package and Instance	
H-HostID *HOST Identification data*	N/A	KDF A2912	N/A	Imported as part of mutual authenticate service's incoming data I/O type: electronic	N/A	Plaintext/dynamic in transient byte array owned by the FIPS Applet inside the Module	Overwrite with all zeros value on: On RESET (Warm/Cold)	Used as FixedInfo data in Key Confirmation and UserToken
OTHER PARAMETERS (not considered as SSPs but included here for completeness)								
H-KT_3DES *Key Token DES* DES Key	TDES-64 TDES-128 TDES-192	CKG	Generated internally using approved CKG (Direct Generation of Symmetric Key) in Generate Key service	Imported via Import Key service and Exported via Export Key service I/O type: electronic	**H-KT_3DES** also can be generated from Diversify Key service using KDF Counter mode with approved MasterKey (**H-KT_AES** or **H-KT_HMAC**)	Plaintext/dynamic as a volatile data inside the System module	Key token data will be zeroised when the user is blocked, System is blocked or terminated, and device reset (warm/cold). Also when the following services are executed:	Needed by some product outside the cryptographic module for compatibility purpose.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroisation	Use & related keys
							- System Reset - Manage session - Sync write token	
OS-MKEK Key Encryption Key AES Key (Not a SSP but list here for completeness)	AES-128	CKG	Generated using approved CKG (Direct Generation of Symmetric Key) function as described in SP800-90	N/A	N/A	Plaintext/static in persistent memory of the Module at manufacturing stage	Key is erased upon Card Manager lifecycle switched to TERMINATED (Overwrite with all zeros)	Master Key used to encrypt (obf uscate) storage of CSPs

Table 16 SSPs

9.2 SSPs Access

SSPs are securely stored in the Cryptographic Module, a secure element hardware that is considered as secure enclave. The SSPs stored in the module is associated and accessible only via approved services as you can see in this section. The approved services itself are associated and accessible to some specific roles as described in the table section 4.3.1.

9.3 Random Bit Generator (RBG)

The RBG source within this module is from entropy source with tittle "ENT (P)" in the Table Approved Algorithm.

Entropy sources	Minimum number of bits of entropy	Details
Hardware-TRNG	Minimum entropy of 2.800831 bits per byte	The Entropy Source is a hardware module inside the CM boundary. The Entropy Source supplies the DRBG with more than 92 bytes Since the entropy source provides a min entropy output of at least 2.800831 bits of min entropy per byte, this is sufficient to obtain 256 bits of security strength

Table 17 Non-Deterministic Random Number Generation Specification

The TRNG (or entropy source) output is used to seed DRBG. The use 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

The approved services that uses DRBG are listed in the section 4.3.2 with link to "DRBG".

9.4 SSP Zeroization

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 16 SSPs) 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 16 SSPs with a correct status or an error status.

10 SELF-TESTS

The module has several self-tests that are triggered at pre-operational (manufacturing stage, power on, or prior to its first use), on demand, conditionally, and periodically.

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.

10.1 Pre-Operational Self-Tests

This section describes pre-operational self-test executed at startup (power on).

10.1.1 Pre-Operational Software/Firmware Integrity Test

The software/firmware integrity is verified using a 16-bit EDC, referred to as CRC-16 hereafter.

Test Target	Description
NVM Integrity	CRC-16 performed over all executable (JavaCard packages) in NVM
ROM Code Integrity	CRC-16 performed over all ROM code

Table 18 Integrity Self-Test Target

10.1.2 Pre-Operational Critical Functions Test

This critical function self-test is performed in every power on before executing integrity self-test.

Test Target	Description
CRC-16	Computes CRC-16 from a fixed message and checks the result (a critical function test)
TRNG	Performs Hardware True Random Number Generator tests
DRBG	Performs a fixed input KAT of CTR_DRBG instantiate and generate functions

Table 19 Critical Function Self-test

10.2 Conditional Self-Tests

The module will automatically trigger specific self-test in some conditions. There are several types of conditional self-tests that are employed by the module described in the following section.

10.2.1 Conditional Cryptographic Algorithm Test

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

Algorithm or Test	Test Properties	Test method	Type	Indicator	Details	Conditions	Coverage	Coverage Notes	Period	Periodic Method
CRC-16	CRC-16 bit	KAT	CAST	Pass: Next test Fail: Module in	CRC	Power On	-	-	2.097.151	On Demand & Auto

Algorithm or Test	Test Properties	Test method	Type	Indicator	Details	Conditions	Coverage	Coverage Notes	Period	Periodic Method
				KillCard State						
AES-CBC	AES, 128-bit,	Covered by CAST on KDF-CMAC	CAST	Pass: Next test Fail: Module in KillCard State	Encrypt	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
AES-CBC	AES, 128-bit,	Covered by CAST on AES ECB	CAST	Pass: Next test Fail: Module in KillCard State	Decrypt	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
AES-ECB	AES, 128-bit, ECB	Covered by CAST on KDF-CMAC	CAST	Pass: Next test Fail: Module in KillCard State	Encrypt	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
AES-ECB	AES, 128-bit, ECB	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Decrypt	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
DRBG	CTR_DRBG AES 256	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
DRBG	CTR_DRBG AES 256	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Instantiate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
DRBG	CTR_DRBG AES 256	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Reseed	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
HMAC	HMAC SHA-1, 128-bit	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto

Algorithm or Test	Test Properties	Test method	Type	Indicator	Details	Conditions	Coverage	Coverage Notes	Period	Periodic Method
SHA2-224		Covered by CAST on SHA2-256 – bit	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
SHA2-256	SHA2-256 – bit	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
SHA2-384		Covered by CAST on SHA2-512 – bit	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
SHA2-512	SHA2-512-bit	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
SHA-3	SHA3-512-bit	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
KDF	KDF, AES CMAC 128 -bit	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Generate	Power On	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
ECDSA	P-224 curve	KAT	CAST	Pass: Next test Fail: Module in KillCard State	signature generation,	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
ECDSA	P-224 curve	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Signature verification	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
RSA PSS	2048-bit RSA-STD PSS with SHA2-256	KAT	CAST	Pass: Next test	signature generation STD	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto

Algorithm or Test	Test Properties	Test method	Type	Indicator	Details	Conditions	Coverage	Coverage Notes	Period	Periodic Method
				Fail: Module in KillCard State						
RSA PSS	2048-bit RSA-CRT PSS with SHA2-256	KAT	CAST	Pass: Next test Fail: Module in KillCard State	signature generation CRT	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
RSA PSS	2048-bit RSA PSS with SHA2-256	KAT	CAST	Pass: Next test Fail: Module in KillCard State	Signature verification	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
RSA PKCS 1 V1.5	2048-bit RSA-STD PSS with SHA2-256	Covered by CAST on RSA-PSS	CAST	Pass: Next test Fail: Module in KillCard State	signature generation STD	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
RSA PKCS 1 V1.5	2048-bit RSA-CRT PSS with SHA2-256	Covered by CAST on RSA-PSS	CAST	Pass: Next test Fail: Module in KillCard State	signature generation CRT	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
RSA PKCS 1 V1.5	2048-bit RSA-PSS with SHA2-256	Covered by CAST on RSA-PSS	CAST	Pass: Next test Fail: Module in KillCard state	Signature verification	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
KAS-ECC	KDF	Covered by CAST on ECDSA, KDF and AES CMAC 128 -bit	CAST	Pass: Next test Fail: Module in KillCard state	Generate	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
KTS	AES-CBC & AES CMAC 128 -bit	Covered by-CAST on AES-CBC and AES-CMAC.	CAST	Pass: Next test Fail: Module in KillCard state	-	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto
AES CMAC	AES CMAC 128 -bit	Covered by CAST on KDF	CAST	Pass: Next test Fail: Module in	Encryption (Verify uses encryption)	First Use	Cert #A2912	IG 10.3.A	2.097.151	On Demand & Auto

Algorithm or Test	Test Properties	Test method	Type	Indicator	Details	Conditions	Coverage	Coverage Notes	Period	Periodic Method
				KillCard state						

Table 20 CM Conditional CAST

10.2.2 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 used as authentication key to calculate the MAC of the loaded firmware.

10.2.3 Conditional Pair-Wise Consistency Test

When the module generating RSA and ECC Key Pair via 'Generate Key Pair' service, the module performs pairwise consistency test using sign 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. On five consecutive errors, the module will enter blocked state. An exit procedure from this state is described in [FQR 401 9097 Ed 4] section 7.3.

10.2.4 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 with calculated KCV, the manual entry test is failed 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 4] section 7.3.

10.3 Periodic Self-Test

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-test described in the section 10.1. 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.

10.4 Operator Initiation of Self-Tests

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

10.5 Error States

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	Power cycle	Muted Device

		Periodic Self-Test Failure		

Table 21 Error States

11 LIFE-CYCLE ASSURANCE

11.1 Installation, Initialization and Startup Procedure

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

1. The installation procedure of the module that supports two physical interface ISO7816 T0 protocol and SPI Protocol are described as follow:
    a. Each ISO7816 physical port defined in section 3.1.1 must be installed correctly by connecting each ISO port (VCC, GND, RST, CLK, IO) of 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 3.1.2 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 2.1.2 are already installed in the manufacturing. The configuration is set to FIPS Certified Product (FCP).
3. In the SYSTEM INITIALIZATION state:
    a. Super User shall create new credential and replace its default credential in the module.
    b. Then, Super User shall create new Administrator role in the module.

    When this procedure is successfully done, the module state is automatically changed to KEY CEREMONY state.
4. In the KEY CEREMONY state:
    a. Administrator shall create new roles for Auditor and three Key Custodians role.
    b. Then, Administrator shall initiate KEY CEREMONY session together with three Key Custodians to import or export CCMK of the module using split knowledge procedure.
    c. After that Administrator shall confirm either to keep the configuration still in FIPS Certified Product (FCP) or not. Please note that if Administrator is answering no, then the module will loose its FIPS Certified status. Customer 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 customer 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 4.3.2 is available for users.
    b. Only applet that is already validated under [NIST.FIPS.140-3] evaluation shall be loaded and installed in the module. Otherwise, the module will loose 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 4] 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 document [FQR 401 9097 Ed 4] for Crypto Officer Role and [FQR 401 9098 Ed 3] for User Role.

11.1.1 Components Version Number Retrieval Procedures

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
  - DGI 'DF52'
    APDU command: 80 CA DF 52
    Hardware version is the first byte of the returned data
- 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. The detail is as follow:
  - Complete OS Firmware Information: '097153'
    - Module Identifier: '09715'
    - Version number: '3'

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'

11.2 Secure Sanitization and Destruction Procedure

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 below step:

1. Under role ‘’AA” (Application Administrator) to set card manager state to TERMINATED.
2. Once, it is terminated, OS-MKEK that is used to encrypt all keys stored in the module is zeroised.
3. Once OS-MKEK is zeroised, all keys stored in the module cannot be retrieved in plaintext form. Event encrypted form, it is difficult to those keys which are stored inside the flash on single chip component which are protected with hard tamper-evident coating on the chip.

12 MITIGATION OF OTHER ATTACKS

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 reaches 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 assure 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.

APPENDIX 1. REFERENCES

A.1 NIST References

Reference	Detail Reference
[NIST.FIPS.140-3]	Security Requirements for Cryptographic Modules
[NIST.FIPS.180-4]	Secure Hash Standard (SHS)
[NIST.FIPS.186-4]	Digital Signature Standard (DSS)
[NIST.FIPS.197]	Advanced Encryption Standard (AES)
[NIST.FIPS.198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[NIST.FIPS.202]	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
[NIST.SP.800-38A]	Recommendation for Block Cipher Modes of Operation: Methods and Techniques
[NIST.SP.800-38B]	Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication
[NIST.SP.800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
[NIST.SP.800-56A.Rev3]	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
[NIST.SP.800-56B.Rev2]	Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography
[NIST.SP.800-56C.Rev2]	Recommendation for Key-Derivation Methods in Key-Establishment Schemes
[NIST.SP.800-63b]	Digital Identity Guidelines
[NIST.SP.800-67.Rev2]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[NIST.SP.800-90A.Rev1]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[NIST.SP.800-90B]	Recommendation for the Entropy Sources Used for Random Bit Generation
[NIST.SP.800-108]	Recommendation for Key Derivation Using Pseudorandom Functions (Revised)
[NIST.SP.800-131A.Rev2]	Transitioning the Use of Cryptographic Algorithms and Key Lengths
[NIST.SP.800-133.Rev2]	Recommendation for Cryptographic Key Generation
[NIST.SP.800-140A]	CMVP Documentation Requirements: CMVP Validation Authority Updates to ISO/IEC 24759
[NIST.SP.800-140B]	CMVP Security Policy Requirements
[NIST.SP.800-140E]	CMVP Approved Authentication Mechanisms

A.2 ISO/IEC References

Reference	Detail Reference
[ISO/IEC 7816-3]	“Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical signal and transmission protocols”. - 2006-11-01 - Third edition

	Reference number: ISO/IEC 7816-3:2006(E)
[ISO/IEC 7816-4]	“Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange." - 2013-04-15 - Third edition Reference number: ISO/IEC 7816-4:2013(E)
[ISO/IEC 19790]	Information technology — Security techniques — Security requirements for cryptographic modules
[ISO/IEC 24759]	Information technology — Security techniques — Test requirements for cryptographic modules

A.3 Global Platform References

Reference	Detail Reference
[GPC_Specification_v2.3]	GlobalPlatform Card Specification Version 2.3.1 Public Release – March 2018 Document Reference: GPC_SPE_034
[GPC_CIC]	GlobalPlatform Card - Common Implementation Configuration Version 2.1 Member Release – July 2018 Document Reference: GPC_GUI_080
[GPC_AMD_D]	GlobalPlatform Card Technology - Secure Channel Protocol '03', Card Specification v2.2 – Amendment D Version 1.1.1 - Public Release July 2014 Document Reference: GPC_SPE_014

A.4 FIPS Applet on RookySE References

Reference	Detail Reference
[FQR 401 9097 Ed 4]	FQR 401 9097 Ed 4 - Rooky Crypto Officer Guidance
[FQR 401 9098 Ed 3]	FQR 401 9098 Ed 3 - Rooky User Guidance
[FQR 401 9187 Ed 2]	FQR 401 9187 Ed 2 - Rooky Delivery, Installation, and Destruction Guidance
[FQR 110 A0A1 Ed 1]	FQR 110 A0A1 Ed 1 - SPI Application Note for ROOKY

APPENDIX 2. ACRONYMS AND DEFINITIONS

Acronym	Definition
FIPS	Federal Information Processing Standard
LKD	Local Key Database
LKD_DP	Local Key Database Data Preparation
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
AdminApp	Admin Application
Enc	Encipher
Dec	Decipher
Sig	Signature
Ver	Verify
JCVM	Java Card Virtual Machine

13 DOCUMENT REVISIONS

Date	Change
1.2	- Update table 3 for CKG and KDF to update description/key size - Update table 16 to add CKG in some key and its used - Update table conditional self-test - Add new table for error states - Add new section 11.2 Secure Sanitization and Destruction Procedure
1.1	Renaming some tables according SP800-140B; typographic errors corrections. Change applet version.
1.0	Initial version