Makina Pty Ltd

OpenFIPS201 v2.0 PIV Applet on NXP P71D600

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

Federal Information Processing Standards Publication 140-3 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a sensitive but unclassified (SBU) environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140-3 program. The National Voluntary Laboratory Accreditation Program (NVLAP) provides accreditation to independent testing labs performing FIPS 140-3 testing; the CMVP validates modules meeting FIPS 140-3 validation. Validated is the term given to a module that is documented and tested against the FIPS 140-3 criteria. More information is available on the CMVP website at: https://csrc.nist.gov/projects/cryptographic-module-validation-program.

This non-proprietary Cryptographic Module Security Policy for the OpenFIPS201 v2.0 PIV Applet on NXP P71D600 provides an overview of the product and a high-level description of how it meets the overall Security Level 2 requirements of FIPS 140-3.

The OpenFIPS201 v2.0 PIV Applet on NXP P71D600 may also be referred to as the “module” in this document.

1.2 Security Levels

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

Table 1: Security Levels

1.3 Additional Information

The Module, validated to FIPS 140-3 overall Level 2, is a single chip module (P71D600 so known as “P71”) implementing the Global Platform operational environment, with a Card Manager (ISD/SSD) and an applet, OpenFIPS201 v2.0 (hereafter referred to as the applet).

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Makina shall have no liability for any error or damage of any kind resulting from the use of this document.

Notices
This document may be freely reproduced and distributed in its entirety without modification.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

This module is composed of the P71D600 Java Card Operating Platform (JCOP) by NXP Semiconductors, running the OpenFIPS201 applet by Makina. OpenFIPS201 is a flexible implementation of the NIST Personal Identity Verification (FIPS 201-3) card interface as per SP800-73-5 and SP800-78-5, providing card and cardholder authentication, as well as digital signature and encryption capabilities.

The module is designed to be used either as a standalone processor embedded in a contact or contact-less smartcard, or as an auxiliary security device attached to a host controller as part of a larger system.

The physical form of the module is depicted in Figures 1 and 2 (to scale); the outline depicts the cryptographic boundary, representing the surface of the chip and the bond pads. The red outline in Figure 3 also depicts the cryptographic boundary.
In production use, the module is delivered to either vendors or end user customers either on film frame carrier (FFC) or various packages such as PDM1.1, NXD6.2, MOB6/10 or HVQFN20 package. The package is outside the cryptographic boundary and thus excluded from the FIPS 140-3/ISO/IEC 19790 security testing.

The contactless ports of the module require connection to an antenna. The module relies on [ISO 7816] and [ISO 14443] card readers as input/output devices, or a [NXP I2C] connection to a host controller.

The Module is composed of a GlobalPlatform operational environment and a single Java Card applet – the OpenFIPS201 applet, running on the NXP P71D600 chip.

The Module has a limited operational environment under the FIPS 140-3 definitions. The Module includes a firmware load function to support necessary updates. New firmware versions within the scope of this validation must be validated through the CMVP. Any other firmware loaded into this Module is out of the scope of this validation and requires separate FIPS 140-3 validation.

The GlobalPlatform operational environment is identified by the following elements:

- The ROM ID
- The Platform ID (a data string that allows the identification of the P71D600 Card Manager component)
- The Patch ID
- Other information describing the content in ROM, NVM and loaded patches.

The OpenFIPS201 Applet is identified by the following elements:

- The Applet Name
- The Applet Version comprised of major, minor and revision parts
- The 'FIPS Mode' status flag denoting operation in the Approved Mode to in turn indicate that the applet build is the FIPS 140-3 validated one.

No components (barring the chip packaging) have been excluded from within the cryptographic boundary.

- The P71D600 GlobalPlatform operational environment component can be identified by using the IDENTIFY APDU command (Info service). This command returns the card identification data, which includes a Platform ID, a Patch ID and other information that allows the identification of the content in ROM, NVM and loaded patches. The Platform ID is a data string that allows the identification of the P71D600 Card Manager component.
- The firmware module (Applet) may be identified using the OpenFIPS201 GET VERSION command

The IDENTIFY APDU command is formatted as follows:

Code	Value	Parameter settings
CLA	'80'	GlobalPlatform
INS	'CA'	GET DATA (IDENTIFY) - ISD
P1	'00'	High order tag value
P2	'FE'	Low order tag value - proprietary data
Lc	'02'	Length of data field
Data	'DF28'	Module identification data
Le	'00'	Length of response data

The command answers the content of the DF28 file:

- Tag 02 identifies the Patch ID
- Tag 03 identifies the Platform Build ID which is made up of the Platform ID (16 Bytes) and the platform build fingerprint (8 Bytes)
- Tag 08 identified the ROM ID.

To verify that the GlobalPlatform operational environment runs in the Approved mode of operation, use the IDENTIFY APDU (as described above).

The DF28 file tag '05' contains the status of the Approved mode compliancy, where '00' identified the Approved mode not active and '01' – Approved mode mode active.

- The OpenFIPS201 Applet may be identified using the OpenFIPS201 GET VERSION command

Code	Value	Parameter settings
CLA	‘00’	ISO 7816 Proprietary Class
INS	‘CB’	PIV GET DATA
P1	‘3F’	High order tag byte
P2	‘00'	Extended (OpenFIPS201) object
Lc	‘05’	Length of data field
Data	‘5C032F4756’	Tag “2F4756” (`/GV`)
Le	‘00’	Length of response data

This command answers with a `GetVersionResponse` object encoded in BER-TLV format:

```
OpenFIPS201_GET_VERSION_Schema DEFINITIONS IMPLICIT TAGS ::=
BEGIN

GetVersionResponse ::= [APPLICATION 19] SEQUENCE {
        application       [0] UTF8String (SIZE(0..127)),
        major             [1] INTEGER (0..127),
        minor             [2] INTEGER (0..127),
        revision          [3] INTEGER (0..127),
        debug             [4] BOOLEAN
}

END
```

Example

```
53 1C
  80 0B 4F70656E46495053323031  -- application = 'OpenFIPS201'
  81 01 02 -- major = 2
  82 01 00 -- minor = 0
  83 01 00 -- revision = 0
  84 01 00 -- debug = false
```

- Status can be retrieved from the OpenFIPS201 Applet using the following command:

Code	Value	Parameter settings
CLA	‘00’	ISO 7816 Proprietary Class
INS	‘CB’	PIV GET DATA

P1	‘3F’	High order tag byte
P2	‘00'	Extended (OpenFIPS201) object
Lc	‘05’	Length of data field
Data	‘5C032F4753’	Tag “2F4753” (`/GS`)
Le	‘00’	Length of response data

This command answers with a `GetStatusResponse` object encoded in BER-TLV format:

```
OpenFIPS201_GET_STATUS_Schema DEFINITIONS IMPLICIT TAGS ::=
BEGIN

AppletState ::= ENUMERATED {
   selectable             (0),
   personalised   (1),
   blocked                (2),
   terminated     (127)
}

GetStatusResponse ::= [APPLICATION 19] SEQUENCE {
   appletState            [0] AppletState,
   operatorRoles          [1] BOOLEAN,
   operatorId             [2] BOOLEAN,
   operatorImmediate      [3] BOOLEAN,
   smState                [4] BOOLEAN,
   vciState               [5] BOOLEAN,
   contactless            [6] BOOLEAN,
   fipsMode               [7] BOOLEAN
}

END
```

Example

```
53 15
  80 01 01 -- appletState = PERSONALISED
  81 01 FF -- operatorRole = ROLE_USER
  82 01 00 -- operatorId = NONE
  83 01 00 -- operatorImmediate = TRUE
  83 01 00 -- smState = FALSE
  85 01 00 -- vciState = FALSE
  86 01 00 -- contactless = FALSE
  87 01 FF -- fipsMode = TRUE
```

Module Type: Hardware

Module Embodiment: Single Chip

Cryptographic Boundary:

The JavaCard and Global Platform APIs are internal interfaces available to the applet. In the Approved mode only the OpenFIPS201 applet and Card Manager (ISD/SSD) services are available at the card edge (the interfaces that cross the cryptographic boundary).

The physical form of the module is depicted in Figures 1 and 2 (to scale); the outline depicts the cryptographic boundary, representing the surface of the chip and the bond pads. The red outline in Figure 3 also depicts the cryptographic boundary.

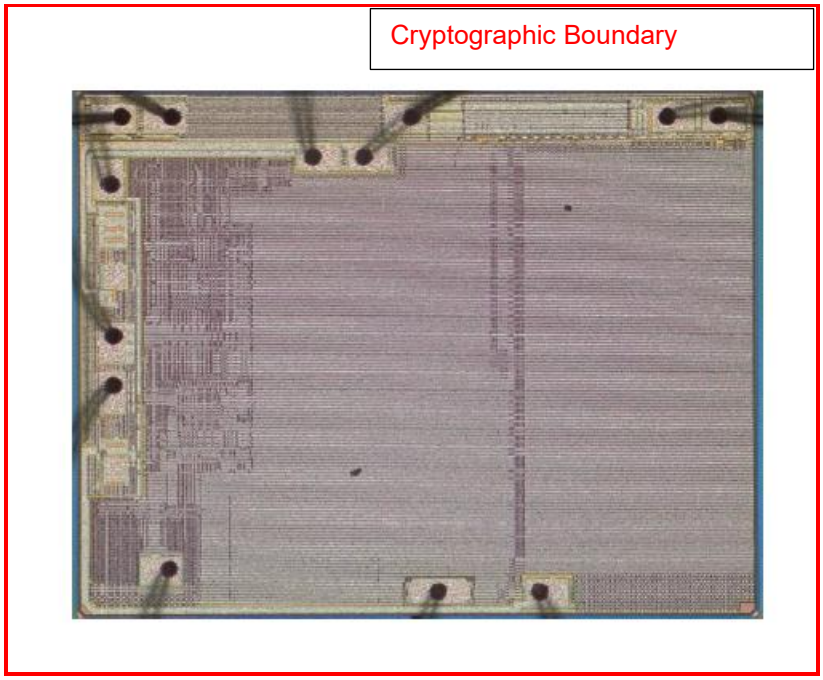

Figure 1: P71D600

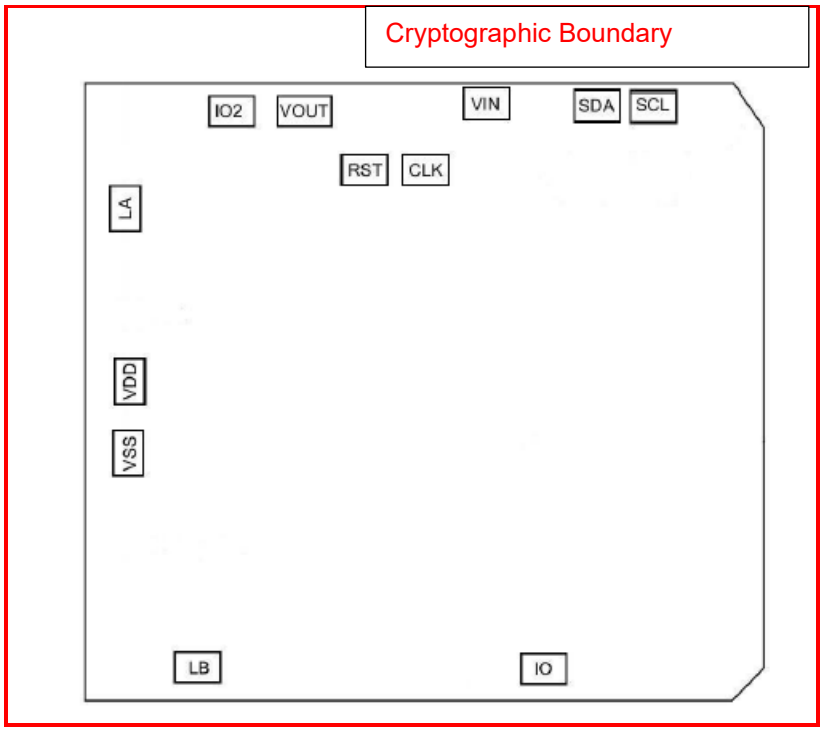

Figure 2: P71D600 Physical form (Schematic)

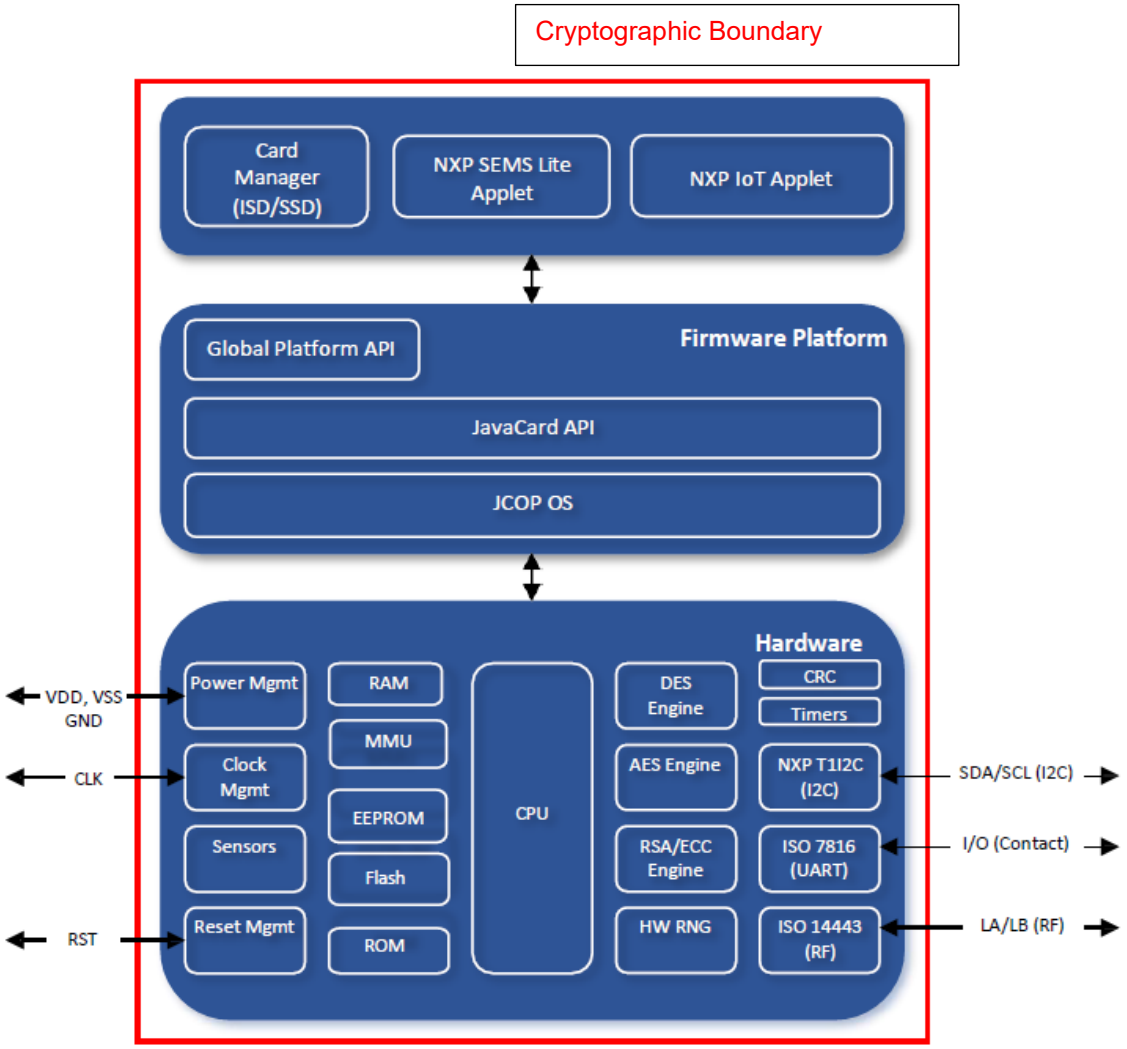

Figure 3: Module Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
P71D600	N7122 A1	Platform ID: J3R6000373181200 ROM ID: B3375FE9B5508BC4 Patch ID: 0000000000000000	MRK3-SC 16/32-bit RISC CPU	The GlobalPlatform operational environment is identified with the Platform ID, the ROM

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
		OpenFIPS201 applet version: Label: OpenFIPS201-P71D600 Major: 2 Minor: 0 Revision: 0 Variant: FIPS		ID, the Patch ID, and other information, describing the content in ROM, NVM and loaded patches; The Platform ID is a data string that allows the identification of the P71D600 Card

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

No components have been excluded from the module boundary (barring the chip packaging).

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	The module supports the Approved mode of operation after following the initialization instructions outlined in Section 11.1	Approved	To verify that the GlobalPlatform operational environment runs in the Approved mode of operation, use the IDENTIFY APDU; The DF28 file tag '05' contains the status of the Approved mode compliancy, where '00' identifies Approved mode not active and '01' - Approved mode active

Table 3: Modes List and Description

Mode Change Instructions and Status:

The module supports an Approved mode of operation by default.

The P71D600 GlobalPlatform operational environment component can be identified by using the IDENTIFY APDU command (Info service). This command returns the card identification data, which includes a Platform ID, a Patch ID and other information that allows the identification of the content in ROM, NVM and loaded patches. The Platform ID is a data string that allows the identification of the P71D600 Card Manager component.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A2713	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A2713	Key Length - 128, 192, 256	SP 800-38C
AES-CMAC	A2713	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A2713	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2713	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
Counter DRBG	A2713	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A6534	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A6534	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A6534	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A2713	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-256	A2713	Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA2-512	A2713	Key Length - Key Length: 256-2048 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A2713	Domain Parameter Generation Methods - P-256 Scheme - onePassDh - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-KC SP800-56 (CVL)	A6718	KAS Role - Responder Key Confirmation Methods - Key Confirmation Directions - Unilateral	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A2713	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-4096 Increment 8 HMAC Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
KDA OneStep SP800-56Cr2	A6718	Derived Key Length - 1024 Shared Secret Length - Shared Secret Length: 224-384 Increment 8	SP 800-56C Rev. 2
KDF SP800-108	A2713	KDF Mode - Counter, Feedback Supported Lengths - Supported Lengths: 112-4096 Increment 8, Supported Lengths: 128	SP 800-108 Rev. 1
RSA Decryption Primitive Sp800-56Br2 (CVL)	A5911	Modulo - 2048, 3072, 4096	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-5)	A6535	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2pow100 Private Key Format - standard	FIPS 186-5
RSA KeyGen (FIPS186-5)	A6536	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2pow100 Private Key Format - crt	FIPS 186-5
RSA SigGen (FIPS186-5)	A6535	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA SigVer (FIPS186-5)	A6535	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
SHA-1	A2713	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A2713	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A2713	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A2713	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG SP800-133r2 Section 4	Key Type:Symmetric and Asymmetric	N/A	NIST SP800-133r2 Section 4: Symmetric keys and seeds for asymmetric key generation are generated using methods described in Section 4 of SP800-133r2 (example 1). The module supports the following per NIST SP 800-133r2: 1. Section 5.1: Key Pairs for Digital Signature Schemes 2. Section 6.2.1: Symmetric Keys Generated Using Key-Agreement Schemes 3. Section 6.2.2: Symmetric Keys Derived from a Pre-existing Key 4.

Name	Properties	Implementation	Reference
			Section 6.4: Distributing the Generated Symmetric Key
CKG SP 800-133r2 Section 6.3	Key Type:Symmetric	N/A	NIST SP 800-133r2 Section 6.3 Symmetric Keys Produced by Combining (Multiple) Keys and Other Data (approach/method 2. Exclusive-Oring one or more symmetric keys and possibly one or more other items of data)

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

The module does not implement any Non-Approved, Allowed Algorithms.

Non-Approved, Allowed Algorithms with No Security Claimed:

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

Non-Approved, Not Allowed Algorithms:

The module does not implement any Non-Approved, Not Allowed Algorithms.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Counter DRBG	DRBG	Random bit generation		Counter DRBG: (A2713)
KAS-1	KAS-Full	NIST SP 800-56Arev3 KAS-ECC per IG D.F Scenario 2 path (2); Used in the context of Secure Messaging	IG:IG D.F Scenario 2 path (2), split Key confirmation:yes Key derivation:KDA (separately tested) Caveat:Key establishment methodology provides 128 bits of security strength	KAS-ECC-SSC Sp800-56Ar3: (A2713) KDA OneStep SP800-56Cr2: (A6718) CKG SP800-133r2 Section 4 : () Key Type: Symmetric and Asymmetric KAS-KC SP800-56: (A6718) SHA2-384: (A2713) SHA2-256: (A2713)

Name	Type	Description	Properties	Algorithms
KAS-2	KAS-Full	NIST SP 800-56Arev3 KAS-ECC per IG D.F Scenario 2 path (2); Used in the context of PIV ECDH Key Agreement	IG :D.F Scenario 2 path (2), split Key confirmation:no Key derivation:KDA (separately tested) Caveat:Key establishment methodology provides 128 bits of security strength	KAS-ECC-SSC Sp800-56Ar3: (A2713) CKG SP800-133r2 Section 4 : () Key Type: Symmetric and Asymmetric
Perform Self-Tests	BC-Auth BC-UnAuth DigSig-SigGen DigSig-SigVer DRBG KAS-56CKDF KAS-SSC KBKDF MAC SHA	Perform self-tests on demand		AES-CBC: (A2713) AES-CMAC: (A2713) Counter DRBG: (A2713) ECDSA SigGen (FIPS186-5): (A6534) ECDSA SigVer (FIPS186-5): (A6534) HMAC-SHA2-256: (A2713) KAS-ECC-SSC Sp800-56Ar3: (A2713) KDA HKDF Sp800-56Cr1: (A2713) KDF SP800-108: (A2713) RSA SigGen (FIPS186-5): (A6535) RSA SigVer (FIPS186-5): (A6535) SHA-1: (A2713) SHA2-256: (A2713) SHA2-512: (A2713)
Asymmetric Key Generation	AsymKeyPair-KeyGen	Aymmetric keypair generation		RSA KeyGen (FIPS186-5): (A6535, A6536)

Name	Type	Description	Properties	Algorithms
				ECDSA KeyGen (FIPS186-5): (A6534) CKG SP800-133r2 Section 4 : () Key Type: Symmetric and Asymmetric
SCP03/secure channel	KBKDF	Sessions keys are used with AES-CBC and AES-CMAC to provide an end-to-end confidential and authenticated protected channel (Approved KTS) between the external entity (User) and the module. KDF SP800-108 Counter is used for deriving the secure channel keys		KDF SP800-108: (A2713) CKG SP800-133r2 Section 4 : () Key Type: Symmetric and Asymmetric
EC Crypto Operations	DigSig-SigGen DigSig-SigVer	ECDSA Signature Generation and Verification		ECDSA SigGen (FIPS186-5): (A6534) ECDSA SigVer (FIPS186-5): (A6534)
RSA Crypto Operations	DigSig-SigGen DigSig-SigVer	RSA Signature Generation and Verification, RSA Decryption		RSA SigGen (FIPS186-5): (A6535) RSA SigVer (FIPS186-5): (A6535) RSA Decryption Primitive Sp800-56Br2: (A5911)
Symmetric Cipher Crypto Operations	BC-Auth BC-UnAuth	Encryption and Decryption		AES-CBC: (A2713) AES-CMAC: (A2713) AES-CCM:

Name	Type	Description	Properties	Algorithms
				(A2713) AES-CTR: (A2713) AES-ECB: (A2713)
MAC Calculation Crypto Operations	MAC SHA	MAC Computation		HMAC-SHA-1: (A2713) HMAC-SHA2-256: (A2713) HMAC-SHA2-512: (A2713) AES-CMAC: (A2713) SHA-1: (A2713) SHA2-256: (A2713) SHA2-512: (A2713)
KTS-1	KTS-Wrap	SP 800-38D and SP 800-38F KTS (key wrapping and unwrapping) per IG D.G	Standard:SP 800-38F IG D.G:approved method from IG D.G Key confirmation:no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A2713) AES-CMAC: (A2713) CKG SP800-133r2 Section 4 : () Key Type: Symmetric and Asymmetric
DAP	DigSig-SigVer SHA	Firmware Load Test		ECDSA SigVer (FIPS186-5): (A6534)
Entropy Source	ENT-ESV	Entropy Input provided by the entropy source within the module to seed the approved NIST SP 800-90Ar1 CTR_DRBG		
OS-MKEK Generation	CKG	OS-SKEK permutation (xor between OS-		CKG SP 800-133r2 Section 6.3: ()

Name	Type	Description	Properties	Algorithms
		SKEK and a constant value)		Key Type: Symmetric

Table 6: Security Function Implementations

2.7 Algorithm Specific Information

IG 2.4.B
RSADP (standard)
Usage Restriction: The RSA Decryption Primitive (CVL) shall only be used within the context of a SP 800-56Brev2 KTS.

Key Confirmation: The key confirmation functionality is used in the context of key agreement (NIST SP 800-56Ar3 KAS-1).

Usage Restriction: The key confirmation functionality by default per the module's design is only used in the context of the NIST SP 800-56Ar3 KAS-1.

The above CVLs have been listed separately in the Section 2.5 Approved Algorithms Table.

IG D.H
Symmetric keys and seeds used for generating the asymmetric keys are generated using methods described in Section 4, Section 5.1, Section 6.2.1, Section 6.2.2 and Section 6.4 of SP 800-133r2.

IG D.L
In the case of the CTR_DRBG, the test report shall indicate if a derivation function is used during the instantiation and reseeding: The CTR_DRBG implementation of the module does use a derivation function (CAVP Cert. #A2713).

In accordance with the Resolution in the IG, the V and Key values for the CTR_DRBG have been defined as CSPs as can be verified from the Section 9.4 SSP Table 1.

IG D.M
Specific requirements for generating symmetric keys using SP 800-108 are found in Sec. 6.4 of SP 800-133rev1, "Symmetric Keys Derived from a Preshared Key." SP 800-108 KDFs may not be used to generate asymmetric keys:

As can be verified from the Section 9.4 SSP Table 2, the module only uses KBKDF to derive symmetric keys.

IG D.F:

KAS-1: Key Agreement i.e. KAS-ECC-SSC per NIST SP 800-56Arev3 combined with a OneStep KDF per NIST SP 800-56Cr2 (KDA) used in the context of the Secure Messaging service (Scenario 2 path (2) per IG D.F – KAS-ECC-SSC and KDA self-tested separately):

KAS (KAS-ECC-SSC CAVP Cert. #A2713 and KDA OneStep NIST SP 800-56Cr2 CAVP Cert. #A6817; P-256 curve providing 128 bits of encryption strength)

KAS-2: Key Agreement i.e. KAS-ECC-SSC per NIST SP 800-56Arev3 combined with an HKDF per NIST SP 800-56Cr1 (KDA) used in the context of PIV ECDH Key Agreement service (Scenario 2 path (2) per IG D.F):

KAS (KAS-ECC-SSC CAVP Cert. #A2713 and KDA HKDF NIST SP 800-56Cr1 CAVP Cert. #A2713; P-256 curve providing 128 bits of encryption strength)

IG D.G:

KTS-1: Key wrapping (Using the approved AES modes CMAC and CBC) used to provide an end-to-end confidential and authenticated protected channel between the external entity (User) and the module (i.e. used in the context of the Secure Channel service). This is per Scenario 2 in IG D.G Approved methods for key transport i.e., a “combination” method: use any approved symmetric encryption mode, such as AES ECB, AES CBC, Triple-DES ECB, etc. together with an approved authentication method (for example, HMAC or AES CMAC, or KMAC):

KTS (AES-CMAC CAVP Cert. #A2713 and AES-CBC CAVP Cert. #A2713; 128, 192, and 256-bit keys providing 128, 192, or 256 bits of encryption strength)

IG 9.6.A
An AES or a Triple-DES encryption using any approved mode of AES or the Triple-DES as defined in SP 800-140C CMVP Approved Security Functions: SSPs are stored encrypted using AES-CBC.

Additional Comment #1: The approved algorithm implementations used to protect stored SSPs shall be tested by the CAVP (or vendor affirmed if allowed by an IG): The AES-CBC has been tested per CAVP Cert#A2713.

IG C.E:
The module generates RSA signature keys using an approved key generation procedure per RSA KeyGen validated for conformance to FIPS 186-5 Cert. #A2713.

IG C.F:
The RSA KeyGen, SigGen and SigVer implementations have been tested for all implemented RSA modulus lengths (moduli 2048, 3072 and 4096 bits). No untested moduli apply.

SHA-1:
The usage of SHA-1 is deprecated through December 31, 2030, for non-digital signature applications and disallowed i.e., non-approved and not allowed thereafter. The module uses SHA-1 for non-digital signature applications, specifically, as the underlying hash function (PRF) for HMAC and NIST SP 800-56Cr1 HKDF.

2.8 RBG and Entropy

Cert Number	Vendor Name
E148	NXP Semiconductors

Table 7: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
JCOP 4.5 on P71D600	Physical	JCOP 4.5 on P71D600	8 bits	7.30359 bits	N/A

Table 8: Entropy Sources

2.9 Key Generation

The module uses an approved NIST SP 800-90Ar1 DRBG for the generation of keys/SSPs.

2.10 Key Establishment

The module supports approved key establishment methods as specified in the Security Functions Implementations tables (KAS-1, KAS-2 and KTS-1).

2.11 Industry Protocols

The module does not support any IETF industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
VSS, VDD	Power	These interfaces are used to supply power to the module in contact mode; The module starts when interface is powered
VIN, VOUT	Power	These interfaces are used to supply power to the module in contact, contactless and I2C mode
RST_N	Control Input	If a signal is sent on this interface on contact mode, the module will reboot (active low)
CLK	Control Input	The interface is used by an external device (ex: smartcard reader) to provide a clock signal to the IC in contact mode; The IC will derive its own clock from this signal
IO1	Data Input Data Output Control Input Status Output	The interface is used to communicate with an external entity (ex: SmartCard reader) in contact mode; It also functions as I2C master SDA in I2C mode
IO2	Data Input Data Output Control Input	The interface is used to communicate with an external entity (ex: SmartCard reader) in contact mode; It also functions as I2C master SCL in I2C mode or as SPI interface

Physical Port	Logical Interface(s)	Data That Passes
	Status Output	
LA, LB	Data Input Data Output Control Input Status Output Power	The interface is used to communicate with an external entity (ex: smartcard reader) in contactless mode; This interface is also used to set the internal clock and to supply power to the module
SDA	Data Input Data Output Control Input Status Output	The interface is used to communicate with an external entity such as a host controller
SCL	Control Input	The interface is used by an external device (ex: host controller) to provide a clock signal to the I2C HW

Table 9: Ports and Interfaces

Note:
1. The module does not support control output and thus does not comprise a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Platform Authentication (Secure Channel Protocol 03 Authentication Method)	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, SD-SMAC,	The external entity participating in the mutual authentication sends a 64-bit challenge to the Secure Element; The Secure Element generates its own challenge and computes a	1/(2^128) = 2.9E39 (MAC||cryptogram) using a 128bit block for authentication; This authentication method includes a counter of failed authentication called "velocity checking" by GlobalPlatform; The counter is decremented prior	The module enforces a maximum of 60 failed Global Platform SCP03 authentication attempts before permanently blocking the card; The probability that a random attempt will succeed over a one-minute interval is (with the assumption here

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	and SD-RMAC session keys; These sessions keys are used with AES-CBC and AES-CMAC to provide an end-to-end confidential and authenticated protected channel (Approved KTS) between the external entity (User) and the module	64-bit cryptogram with SD-SMAC key and both challenges; The Secure Element cryptogram and challenge are sent to the external entity which checks the Secure Element 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 external entity cryptogram with AES-CMAC and SD-SMAC key; The MAC is concatenated to the command, and the command is sent to the Secure Element; The Secure Element	to any attempt to authenticate and is only reset to its threshold (maximum value) upon successful authentication	that one attempt is possible per second): 60/(2^128) = 1.7E-37 (MAC||cryptogram), using a 128bit block for authentication

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
		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)		
PIV Symmetric External (AUTH_SYM_EXT)	The Key Holder is authenticated by the Application using a two-stage challenge / response protocol, provided by the PIV CM Authentication service using the AES-AUTH-EXT CSP	1. A 128-bit challenge is retrieved from the Application 2. The external entity encrypts the challenge and returns it to the Applet with a reference to associated key and mechanism 3. The Applet decrypts the challenge response with the referenced key and mechanism; If the decrypted challenge response	The strength of this authentication method depends on the type and length of key referenced; The worst-case scenario is an AES key with 128-bit key length and a block (challenge) size of also 128-bits; The probability that a random attempt will succeed using this authentication method is: 1/(2^128) = 2.9E39	This authentication method is inherently rate-limited by the performance of the underlying platform and communications protocol; The probability that a random attempt will succeed over a one-minute interval is (with the assumption here that 10 attempts are possible per second): 600/(2^128) = 1.7E-36

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
		matches the original challenge, the Applet sets the appropriate internal security status indicator 4. The Applet returns an indicator of whether the authenticatio n succeeded or failed to the external entity 5. The external entity may request the current authenticatio n security status from the Applet via the GET STATUS command at any time		
PIV Symmetric Mutual (AUTH_SYM_MUT)	The Application Administrator (AA) and the Applet authenticate to each-other by using a three-stage challenge / response protocol	1. A 128-bit random witness generated and encrypted by the Application, then returned to the external entity 2. The external entity encrypts the witness to produce the witness	The strength of this authentication method depends on the type and length of key referenced; The worst-case scenario is an AES key with 128-bit key length and a block (challenge) size of also 128-bits; The probability that a random attempt will succeed using this authentication method is:	This authentication method is inherently rate-limited by the performance of the underlying platform and communications protocol; The probability that a random attempt will succeed over a one-minute interval is (with the assumption here that 10 attempts are possible per second):

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
		response and returns this to the Application with a reference to associated key and mechanism, along with an externally generated 128-bit challenge 3. The application decrypts the witness response with the referenced key and mechanism and compares it to the original witness value, if this is an exact match, the Application sets the appropriate internal authenticatio n status and authenticatio n continues 4. The Application encrypts the challenge with the referenced key and mechanism and returns the challenge response,	1/(2^128) = 2.9E39	600/(2^128) = 1.7E-36

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
		along with a status to indicate authentication has succeeded 5. If the decrypted challenge response matches the original challenge, the external entity sets the appropriate internal security status indicator 6. The external entity may request the current authentication security status from the Application via the GET STATUS command at any time		
PIV Local PIN Verification (AUTH_L_PIN)	The User role authenticates to the Application by using the application-held (local) PIN; The PIN value must be explicitly set by the SO or AA roles, before the Module is	N/A	In the worst-case NPIVP compliant mode, the Module will accept a 6-byte PIN value comprised of numeric digits 0 to 9 (byte values 30h to 39h respectively), giving 10 possible values per byte; In addition to this, all PIN values are permitted to be	The module also enforces a failed attempt limit (retries), which can be configured up to a maximum 15 attempts; After this many retries, the PIN will automatically be blocked until the SO or AA role unblocks and resets it; The probability that a

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	delivered to the User		configured up to 16-byte lengths, with an optional padding byte FF used for shorter PIN values; Regardless of the length of the PIN, all the bytes are compared to the stored PIN reference value; The probability that a random attempt will succeed using this authentication method is: 1 / (10 ^ 6) = 1E-6	random attempt will succeed using this authentication method over a one-minute interval is: 15 / (10 ^ 6) = 1.5E-5
PIV Global PIN Verification (AUTH_G_PIN)	The User role authenticates to the Application by using the card-held (global) PIN; The PIN value must be explicitly set by the CO role, before the Module is delivered to the User	N/A	In the worst-case NPIVP compliant mode, the Module will accept a 6-byte PIN value comprised of numeric digits 0 to 9 (byte values 30h to 39h respectively), giving 10 possible values per byte; In addition to this, all PIN values are permitted to be configured up to 16-byte lengths, with an optional padding byte FFh used for shorter PIN values. Regardless of the length of the PIN, all the bytes are compared to the stored PIN reference value; The probability that a random attempt will	The module also enforces a failed attempt limit (retries), which can be configured up to a maximum 15 attempts; After this many retries, the PIN will automatically be blocked until the SO or AA role unblocks and resets it; The probability that a random attempt will succeed using this authentication method over a one-minute interval is: 15 / (10 ^ 6) = 1.5E-5

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
			succeed using this authentication method is: 1 / (10 ^ 6) = 1E-6	
PIV Security Officer Verification (AUTH_PUK)	The Security Officer (SO) role authenticates to the card using the PUK (PIV-PUK) to permit the operation of resetting the PIV-PIN-LOCAL or the PUK itself	N/A	In the worst-case NPIVP compliant mode, the Module will accept a 6-byte PUK value comprised of any possible value, giving 256 possible values per byte; The probability that a random attempt will succeed using this authentication method is: 1 / (10 ^ 6) = 1E- 6	The module also enforces a failed attempt limit (retries), which can be configured up to a maximum 15 attempts; After this many retries, the PUK will automatically be blocked until the AA role unblocks and resets it; The probability that a random attempt will succeed using this authentication method over a one-minute interval is: 15 / (10 ^ 6) = 1.5E-5

Table 10: Authentication Methods

The module employs (and enforces i.e. the method is required) identity-based authentication mechanisms.

Authentication of each operator and their access to roles and services is as described below, independent of logical channel usage.
• The SCP03 authentication method is the only method available to the operator on first use. This permits the Cryptographic Officer (CO) to configure the module.
• Only one operator at a time is permitted on a channel.
• Applet de-selection (including Card Manager), card reset, or power down terminates the current authentication. Re-authentication is required after any of these events for access to authenticated services.
• CO authentication method does not exchange plaintext CSP.
• User authentication data is encrypted and authenticated during entry with GlobalPlatform SCP03. The PINs and PUK can also be entered in plaintext over the contact interface. All authentication data is stored encrypted with OS-MKEK and is only accessible by authenticated services.
• Each authentication method for authenticating to the OpenFIPS201 applet has an explicitly defined scope, that is to say a lifetime outside of which the authentication is reset automatically and the operator is required to re-authenticate again in order to access authenticated services.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Cryptographic Officer (CO)	Identity	Cryptographic Officer	Platform Authentication (Secure Channel Protocol 03 Authentication Method)
User	Identity	User	Platform Authentication (Secure Channel Protocol 03 Authentication Method)
Card Holder	Identity	User	PIV Local PIN Verification (AUTH_L_PIN) PIV Global PIN Verification (AUTH_G_PIN)
Security Officer (SO)	Identity	Cryptographic Officer	PIV Security Officer Verification (AUTH_PUK)
Key Holder	Identity	Cryptographic Officer	PIV Symmetric External (AUTH_SYM_EXT) PIV Symmetric Mutual (AUTH_SYM_MUT)
Application Administrator (AA)	Identity	Cryptographic Officer	Platform Authentication (Secure Channel Protocol 03 Authentication Method)

Table 11: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Manage Content (ISD Services)	Load keys and data; APDU(s) used: DELETE, LOAD, INSTALL, MANAGE CHANNEL	Status Word (Response APDU 9000)	Command parameters (data objects, SSPs)	Status Word (Response APDU 9000)	KTS-1 DAP	Cryptographic Officer (CO) - OS-SKEK: W,E,Z - SD-KENC: W,E,Z - SD-KMAC: W,E,Z - SD-KDEK: W,E,Z - DAP-DAPK: W,E
Lifecycle (Show status and Perform zeroisation	Get or modify the card or applet life cycle status; APDU(s) used: SET	Status Word (Response APDU 9000)	Target status	Status Word (Response APDU 9000)	DAP	Cryptographic Officer (CO) - OS-DRBG-EI: E,Z - OS-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
) (SSD Services)	STATUS, GET STATUS, TERMINATE (Zeroise)					DRBG- KEY: E,Z - OS-DRBG-V: E,Z - OS-SKEK: E,Z - OS-MKEK: E,Z - SD-KENC: E,Z - SD-KMAC: E,Z - SD-KDEK: E,Z - SD-SENC: E,Z - SD-SMAC: E,Z - SD-RMAC: E,Z - DAP-DAPK: E,Z - RSA-PRIV-KEY: E,Z - EC-PRIV-KEY: E,Z - EC-KAS-Z: E,Z - AES-AUTH-INT: E,Z - AES-AUTH-EXT: E,Z - AUTH-PIN-L: E,Z - AUTH-PIN-G: E,Z - AUTH-PC: E,Z - AUTH-PUK: E,Z - AUTH-PIN-HIST: E,Z - EC-SM-Z : E,Z - EC-SM-SKENC: E,Z - EC-SM-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						SKCMAC: E,Z - EC-SM-SKRMAC: E,Z - AES-AUTH-RND1: E,Z - AES-AUTH-RND2: E,Z - RSA-PUB-KEY: E,Z - EC-PUB-KEY: E,Z - RSA-CERT: E,Z - EC-CERT: E,Z - EC-KAS-PUB: E,Z - EC-SM-CVC: E,Z
Manage Content (SSD Services)	Load keys and data; APDU(s) used: PUT KEY, STORE DATA	Status Word (Response APDU 9000)	Command parameters (data objects, SSPs)	Status Word (Response APDU 9000)	KTS-1 DAP	Cryptographic Officer (CO) - OS-SKEK: W,E,Z - SD-KENC: W,E,Z - SD-KMAC: W,E,Z - SD-KDEK: W,E,Z - DAP-DAPK: W,E
Privileged Info (Show module's versioning information) (SSD Services)	Read Module data (privileged data objects, but no CSPs); APDU(s) used: GET DATA	Status Word (Response APDU 9000)	Command parameters (privileged data objects, but no SSPs)	Requested information; Status Word (Response APDU 9000)	None	Cryptographic Officer (CO)
Secure Channel	Establish and use a secure communicati	Status Word (Response	Command paramet	Status Word (Respon	Counter DRBG SCP03/sec	Cryptographic Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
(SSD Services)	on channel; APDU(s) used: INITIALIZE, UPDATE, EXTERNAL AUTHENTICATE	APDU 9000)	ers (data objects, SSPs)	se APDU 9000)	ure channel KTS-1 Entropy Source OS-MKEK Generation	- OS-DRBG-EI: G,E,Z - OS-DRBG- KEY: G,E,Z - OS-DRBG-V: G,E,Z - OS-MKEK: G,E,Z - SD-KENC: G,E,Z - SD-KMAC: G,E,Z - SD-SENC: G,E,Z - SD-SMAC: G,E,Z - SD-RMAC: G,E,Z - OS-DRBG-SEED: G,E,Z
Card Reset	Power cycle or reset the module; APDU(s) used: N/A	Status Word (Response APDU 9000)	Power cycle or reset the module	Status Word (Response APDU 9000)	None	Cryptographic Officer (CO) User Unauthenticated
Context	Select an applet or manage logical channels; APDU(s) used: SELECT, MANAGE CHANNEL	Status Word (Response APDU 9000)	Command parameters (data objects, SSPs)	Status Word (Response APDU 9000)	None	Cryptographic Officer (CO) User Unauthenticated
Info (Show status and Perform self-tests)	Read unprivileged data objects, e.g., module configuration or status information (Show Status); This service	Status Word (Response APDU 9000)	Command parameters (data objects, SSPs)	Status Word (Response APDU 9000)	Perform Self-Tests	Cryptographic Officer (CO) User Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	includes the Pre-operational Self-Test on-demand; APDU(s) used: GET DATA					
PIV Key Generation	Generates asymmetric key-pairs on the card and returns the public component in accordance with [800- 73-4]; APDU(s) used: PIV GENERATE ASYMMETRIC KEYPAIR	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier	Key Public Data, Status Word (Response APDU 9000)	Counter DRBG Asymmetric Key Generation Entropy Source	Cryptographic Officer (CO) - RSA-PRIV-KEY: G - EC-PRIV-KEY: G - RSA-PUB-KEY: G,R - EC-PUB-KEY: G,R Application Administrator (AA) - RSA-PRIV-KEY: G - EC-PRIV-KEY: G - RSA-PUB-KEY: G,R - EC-PUB-KEY: G,R User - RSA-PRIV-KEY: G - RSA-PUB-KEY: G,R - EC-PRIV-KEY: G - EC-PUB-KEY: G,R
PIV Digital Signature	Signs an externally generated preformatted signature block in accordance with [800-73-4]; APDU(s) used: PIV	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier, Pre-Formatted Signature Block	Signed Signature Block, Status Word (Response APDU 9000)	EC Crypto Operations RSA Crypto Operations	User - RSA-PRIV-KEY: E - EC-PRIV-KEY: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	GENERAL AUTHENTICATE					
PIV RSA Key Transport	Decrypts an externally generated, formatted, and encrypted RSA block containing a pre-master secret in accordance with [800-73-4] and [800-56B]; APDU(s) used: PIV GENERAL AUTHENTICATE	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier, Encrypted Pre-Master Secret	Decrypted Pre-Master Secret, Status Word (Response APDU 9000)	RSA Crypto Operations	User - RSA-PUB-KEY: E
PIV ECDH Key Agreement	Derives a pre-master secret based on an externally supplied ECC public key and an on-card ECC private key in accordance with [800-73-4] and [800-56A]; APDU(s) used: PIV GENERAL AUTHENTICATE	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier, Host ECC public key	Shared Secret Z, Status Word (Response APDU 9000)	KAS-2	User - EC-KAS-Z: G,E,Z - EC-KAS-PUB: W,E,Z - EC-PRIV-KEY: G,E,Z
PIV Card Authentication	Authenticates the Card to the host system in accordance with [800-73-4]; APDU(s)	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier, Host	Encrypted Challenge Response, Status Word	None	Cryptographic Officer (CO) - AES-AUTH-EXT: W,E User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	used: PIV GENERAL AUTHENTICATE		Challenge Data	(Response APDU 9000)		- AES-AUTH-EXT: W,E Unauthenticated Card Holder - AES-AUTH-EXT: W,E Security Officer (SO) - AES-AUTH-EXT: W,E Key Holder - AES-AUTH-EXT: W,E Application Administrator (AA) - AES-AUTH-EXT: W,E
PIV Secure Messaging	Establishes a secure channel to provide a protected plaintext channel between the host system and the card; Also permits binding of the terminal / host system to the card; APDU(s) used: PIV GENERAL AUTHENTICATE	Status Word (Response APDU 9000)	Key identifier, Mechanism identifier, Host ECC public key	Session key, Status Word (Response APDU 9000)	KAS-1 Symmetric Cipher Crypto Operations MAC Calculation Crypto Operations	User - EC-SM-Z : G,E,Z - AUTH-PC: W,E - EC-SM-SKENC: W,E - EC-SM-SKCMAC: W,E - EC-SM-SKRMAC: W,E - EC-KAS-PUB: W,E,Z - EC-SM-CVC: W,E Unauthenticated - EC-SM-Z : G,E,Z - EC-SM-SKENC: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- EC-SM-SKCMAC: W,E - EC-SM-SKRMAC: W,E - EC-KAS-PUB: W,E,Z - EC-SM-CVC: W,E
Manage Card OS	-	Status Word (Response APDU 9000)	-	Status Word (Response APDU 9000)	None	Cryptographic Officer (CO)
Manage Card Content	APDU(s) used: GP DELETE, GP LOAD, GP INSTALL, GP MANAGE CHANNEL, GP PUT KEY, GP STORE DATA	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	Cryptographic Officer (CO)
Read Card Privileged Info	APDU(s) used: GP GET DATA	Status Word (Response APDU 9000)	Command Parameters	Privileged (nonCSP) Info, Status Word (Response APDU 9000)	None	Cryptographic Officer (CO)
Manage Card Lifecycle	APDU(s) used: GP SET STATUS, GP GET STATUS	Status Word (Response APDU 9000)	Command Parameters	Card Status, Status Word (Response APDU 9000)	None	Cryptographic Officer (CO)
Manage Secure Channel	APDU(s) used: GP INITIALIZE UPDATE, GP EXTERNAL AUTHENTICATE	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	SCP03/secure channel KTS-1	Cryptographic Officer (CO) - OS-DRBG-EI: G,E,Z - OS-DRBG- KEY:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,E,Z - OS-DRBG-V: G,E,Z - OS-MKEK: G,E,Z - SD-KENC: G,E,Z - SD-KMAC: G,E,Z - SD-SENC: G,E,Z - SD-SMAC: G,E,Z - SD-RMAC: G,E,Z - OS-DRBG-SEED: G,E,Z
Manage Application Data	Permits management of the application data object contents; APDU(s) used: PUT DATA	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	Cryptographic Officer (CO) Application Administrator (AA) User
Manage Application Configuration	Permits management of the application file system (data object and key containers) and configuration; Can also inject private and public key values; APDU(s) used: PUT DATA ADMIN, CHANGE REFERENCE DATA ADMIN	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	Application Administrator (AA) - RSA-PRIV-KEY: W - EC-PRIV-KEY: W - RSA-PUB-KEY: W - EC-PUB-KEY: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Manage Verification Data	Permits management of cardholder verification reference data (PIN and PUK values) by either the owning party or an authorised party; APDU(s) used: CHANGE REFERENCE DATA, CHANGE REFERENCE DATA ADMIN	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	User - AUTH-PIN-L: R,W - AUTH-PIN-G: R,W - AUTH-PUK: R,W - AUTH-PIN-HIST: R,W Security Officer (SO) - AUTH-PIN-L: R,W - AUTH-PIN-G: R,W - AUTH-PUK: R,W - AUTH-PIN-HIST: R,W Application Administrator (AA) - AUTH-PIN-L: R,W - AUTH-PIN-G: R,W - AUTH-PUK: R,W - AUTH-PIN-HIST: R,W
Unblock Verification Data	Permits unblocking and changing of the Cardholder Verification data (PIN) by the SO role on presentation of the correct PUK value; APDU(s) used: RESET RETRY COUNTER	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	Security Officer (SO) - AUTH-PIN-L: W - AUTH-PIN-G: W - AUTH-PUK: W,E - AUTH-PIN-HIST: W
Applet Selection	APDU(s) used: SELECT	Status Word (Response	Command	Status Word (Respon	None	Cryptographic Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		APDU 9000)	Parameters	se APDU 9000)		User Card Holder Security Officer (SO) Key Holder Application Administrator (AA) Unauthenticated
Reset Verification Status	De-authorises the User role on request; APDU(s) used: PIV VERIFY	Status Word (Response APDU 9000)	Command Parameters	Status Word (Response APDU 9000)	None	User
Get Extended Application Data (Status and Applet Version Reporting)	APDU(s) used: GET DATA EXTENDED	Status Word (Response APDU 9000)	None	Application Status, Application Version, Data Object, Key and PIN information (no content, just header), Random Data, Status Word (Response APDU 9000)	None	Cryptographic Officer (CO) User Card Holder Security Officer (SO) Key Holder Application Administrator (AA) Unauthenticated
Terminal Binding	A device authenticates to the Application using an 8-character numeric value; This	Successful authentication of the module to the endpoint	Pairing Code (8-character numeric value)	-	None	User - AUTH-PC: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	can only be performed once the PIV Secure Messaging service has been used to establish a protected plaintext channel between the Device and the Application					
Data Object Reading	The role required to read depends on the access control permissions specified for each data object; APDU(s) used: GET DATA	Status Word (Response APDU 9000)	Data Object Identifier	Data Object Content, Status Word (Response APDU 9000)	None	User

Table 12: Approved Services

The modes of access shown in the table above are defined as:
• G = Generate: The service generates or derives the CSP/Public Key.
• W = Write: The service inputs the CSP/Public Key.
• E = Execute: The Module executes using the CSP/Public Key.
• R = Read: The service outputs the CSP/Public Key. CSP are always protected with the approved KTS.
• Z = Zeroise: The Module zeroises the CSP/Public Key after usage. A zeroised CSP is not retrievable or reusable.

4.4 Non-Approved Services

The module does not support any Non-Approved Services.

4.5 External Software/Firmware Loaded

The module contains a limited operational environment. The module supports loading of firmware from an external source, namely the OpenFIPS201 Applet and performs a firmware

load test (using ECDSA P-256 SHA2-256) in support of the partial load. The firmware load test fails without loading the reference authentication key i.e. the ECDSA P-256 SHA2-256 public key prior to the load.

5 Software/Firmware Security

5.1 Integrity Techniques

The cryptographic module is considered a hardware module with firmware components. An error detection code (32-bit CRC performed over all code located in Flash) is applied to all firmware components within the module. If the integrity test fails, the module enters the hard error (MUTE) state. The module is a single-chip and the executable code of the module (firmware), is Register Transfer Language (RTL). All data and control inputs, and data, control and status outputs of the cryptographic module and services are directed through the module's defined interfaces.

5.2 Initiate on Demand

An operator of the module can perform the integrity test on demand with the GET DATA APDU command.

5.3 Additional Information

ROM endurance has been proven to be more than 10 years after manufactured date. Therefore, per FIPS 140-3 IG 5.A, no pre-operational ROM integrity self-test has been implemented. The module's end-of-life procedures must be applied prior to the degradation of the ROM by setting the module to the TERMINATE state.

All data and control inputs, and data and status outputs of the cryptographic module and services are directed through the module's defined interfaces.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

The module claims to meet Physical Security Level 4 and thus the requirements per this section do not apply.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Hard, tamper evident coating	N/A	N/A

Table 13: Mechanisms and Actions Required

The module is a single-chip implementation that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. The module uses standard passivation techniques. The module includes Environmental Failure Protection features such as temperature and voltage sensors. Fault Induction mitigation techniques are light sensors and spike sensors on the supply voltage lines.

Identification of internal features such as sensitive components or interconnections is impeded by a fine mesh of metal shield lines that resides at the outermost layers of the chip.

Delivery forms of the module are QFN package, contactless chip card module, or sawn wafer. Therefore, the module does not rely on any physical security based on a package.

7.2 Fault Induction Mitigation

As specified in Section 12 of this document, the module implements fault induction mitigations such as light sensors, voltage glitch sensors and an active shield, and detection of illegal address or instruction.

7.3 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	-40C	EFP	Shutdown
HighTemperature	+105C	EFP	Shutdown
LowVoltage	1.62V	EFP	Shutdown
HighVoltage	6.0V	EFP	Shutdown

Table 14: EFP/EFT Information

7.4 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	-45C
HighTemperature	+125C

Table 15: Hardness Testing Temperatures

8 Non-Invasive Security

8.1 Mitigation Techniques

In accordance with FIPS 140-3 IG 12.A Additional Comment 1, the non-invasive security mitigations have been specified in Section 12 of this document.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Temporarily stored in RAM	plaintext (does not persist beyond a power cycle); object identifier to entity association	Dynamic
Stored in NVM	plaintext; object identifier to entity association	Static
Stored in NVM - Encrypted	encrypted with Approved AES CBC with OSMKEK; key version to entity association	Static

Table 16: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Exported using Approved KTS	Stored in NVM - Encrypted	External endpoint	Encrypted	Automated	Electronic	KTS-1
Imported in secure channel specified by GP-Amd-I	External endpoint	Stored in NVM - Encrypted	Encrypted	Automated	Electronic	KTS-1
Encrypted using AES-CBC (using SD-KDEK) (RFC 3394 method) and transported using platform SCP03	External endpoint	Stored in NVM encrypted with Approved AES CBC with OS-MKEK	Encrypted	Automated	Electronic	KTS-1
Entered encrypted with the	External endpoint	Stored in NVM - Encrypted	Encrypted	Automated	Electronic	KTS-1

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
previous SD-KDEK						
Not Imported	N/A	N/A	Plaintext	N/A	N/A	
Not Exported	N/A	N/A	Plaintext	N/A	N/A	
Entered in Plaintext	External Endpoint	Temporarily stored in RAM	Plaintext	Manual	Electronic	
Exported in plaintext	Stored in NVM - Encrypted	External endpoint	Plaintext	Manual	Electronic	
Entered during manufacturing/ personalization	Vendor generated at manufacture/during personalization	Stored in NVM - Encrypted	Encrypted	N/A	N/A	
Entered during manufacturing/ personalization - plaintext storage	Vendor generated at manufacture/during personalization	Stored in NVM	Plaintext	N/A	N/A	

Table 17: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Power-off	Ephemeral parameter	Cleared upon power cycling the module	Operator initiation
Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	Overwritten with zeroes	Setting the module to the TERMINATE state triggers zeroisation and renders the module unusable	Operator initiation
Destroyed because of OS-MKEK zeroisation	-	Zeroising the root of trust causes zeroisation of SSPs derived from it	Module initiation
Immediately after output	Cleared after use	Automated zeroisation per module design	Module initiation
Cleared immediately following derivation of EC-SM-SKENC, EC-SM-SKCMAC and EC-SM-SKRMAC	Cleared after use	Automated zeroisation per module design	Module initiation

Zeroization Method	Description	Rationale	Operator Initiation
Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session	Ephemeral session-specific parameter	Terminated on session closure	Module initiation
Cleared upon receipt of client response	Ephemeral parameter	Cleared upon receipt of client response	Module initiation
Cleared on subsequent authentication command	Ephemeral parameter	Cleared upon receipt of subsequent authentication command	Module initiation
Cleared upon receipt of challenge response	Ephemeral parameter	Cleared upon receipt of challenge response	Module initiation

Table 18: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
OS-DRBG-EI	Random value from ENT (P) used to seed the AES-256 CTR_DRBG	384 bits - 384 bits	Entropy Input - CSP	Entropy Source		Counter DRBG
OS-DRBG-SEED	Seed provided to the CTR_DRBG	384 bits - 384 bits	DRBG seed - CSP	Counter DRBG		Counter DRBG
OS-DRBG-KEY	Current DRBG state value (per IG D.L Resolution 3.)	256 bits - 256 bits	DRBG State Value - CSP	Counter DRBG		Counter DRBG
OS-DRBG-V	Current DRBG state value (per IG D.L Resolution 3.)	256 bits - 256 bits	DRBG State Value - CSP	Counter DRBG		Counter DRBG
OS-SKEK	Used to build OS-MKEK	128 bits - 128 bits	Symmetric key - CSP			OS-MKEK Generation
OS-MKEK	Used to encrypt all secret and private key data stored in NVM	128 bits - 128 bits	Symmetric key - CSP	OS-MKEK Generation		Symmetric Cipher Crypto Operations
SD-KENC	Used to derive SD-SENC	128 bits - 128 bits	Symmetric key - CSP			SCP03/secure channel

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SD-KMAC	Used to derive SD-SMAC, SD-RMAC	128 bits - 128 bits	Symmetric key - CSP			Symmetric Cipher Crypto Operations
SD-KDEK	Sensitive data decryption key used to decrypt CSPs	128 bits - 128 bits	Symmetric key - CSP			Symmetric Cipher Crypto Operations KTS-1
SD-SENC	Session encryption key used to secure channel data	128 bits - 128 bits	Symmetric key - CSP		SCP03/secure channel	Symmetric Cipher Crypto Operations
SD-SMAC	Session MAC key used to verify inbound secure channel data integrity	128 bits - 128 bits	Symmetric key - CSP		SCP03/secure channel	Symmetric Cipher Crypto Operations
SD-RMAC	Session MAC key used to verify outbound secure channel data integrity	128 bits - 128 bits	Symmetric key - CSP		SCP03/secure channel	Symmetric Cipher Crypto Operations
DAP-DAPK	ECC public key used for Mandated DAP	P-256 - 256 bits	Public key - Neither			EC Crypto Operations
RSA-PRIV-KEY	Used for performing RSA cryptographic operations (digital signature, key transport)	2048, 3072, 4096 bits - 112, 128, 152 bits	Private key - CSP	Asymmetric Key Generation		RSA Crypto Operations
EC-PRIV-KEY	Static EC key for performing ECDSA and ECDH KAS operations	P-256, P-384 - 128, 192 bits	Private key - CSP	Asymmetric Key Generation		KAS-1 KAS-2 EC Crypto Operations
EC-KAS-Z	Ephemeral EC CDH shared secret (Z) value generated as an input to a key agreement (KAS) operation	P-256, P-384 - 128, 192 bits	Shared Secret - CSP		KAS-2	KAS-2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES-AUTH-INT	AES key used for internal (card) authentication operations (encrypting)	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			
AES-AUTH-EXT	AES key used for external (host) authentication methods (AUTH_SYM_EXT and AUTH_SYM_MUT)	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			
AUTH-PIN-L	Card Holder Local (Application) PIN	N/A - 6 to 16 decimal digits	PIN - CSP			
AUTH-PIN-G	Card Holder Global (Card) PIN	N/A - 6 to 16 decimal digits	PIN - CSP			
AUTH-PC	Secure Messaging - Pairing Code	N/A - 6 to 16 decimal digits	PIN - CSP			
AUTH-PUK	PIN Unblocking Key	N/A - 6 to 16 decimal digits	PIN Unblocking Key - CSP			
AUTH-PIN-HIST	PIN History Values - 1 to 12	N/A - 4 to 16 decimal digits	PIN History Values - CSP			
EC-SM-Z	Secure Messaging - ECDH Shared Secret (Z)	P-256, P-384 - 128 to 192 bits	Shared Secret - CSP		KAS-1	KAS-1
EC-SM-SKENC	Secure Messaging - Session Encryption Key (skENC)	128, 192, 256 bits - 128 to 192 bits	Private key - CSP		KAS-1	Symmetric Cipher Crypto Operations
EC-SM-SKCMAC	Secure Messaging - Session	128, 192, 256 bits	Private key - CSP		KAS-1	MAC Calculation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	Command MAC Key (skCMAC)	- 128 to 192 bits				Crypto Operations
EC-SM-SKRMAC	Secure Messaging - Session Response MAC Key (skRMAC)	128, 192, 256 bits - 128 to192 bits	Private key - CSP		KAS-1	MAC Calculation Crypto Operations
AES-AUTH-RND1	128-bit 'challenge' nonce generated by module as part of the AUTH_SYM_EXT and AUTH_SYM_MUT authentication methods	- - 128 bits	Authentication nonce - Neither	Counter DRBG		Symmetric Cipher Crypto Operations
AES-AUTH-RND2	128-bit 'witness' nonce input into module as part of the AUTH_SYM_EXT and AUTH_SYM_MUT authentication methods	- - 128 bits	Authentication nonce - Neither	Counter DRBG		Symmetric Cipher Crypto Operations
RSA-PUB-KEY	Used for performing RSA cryptographic operations (digital signature, key transport)	2048, 3072, 4096 bits - 112, 128, 152 bits	Public key - PSP	Asymmetric Key Generation		RSA Crypto Operations
EC-PUB-KEY	Static EC key for performing ECDSA and ECDH KAS operations	P-256, P-384 - 128, 192 bits	Public key - PSP	Asymmetric Key Generation		KAS-1 KAS-2 EC Crypto Operations
RSA-CERT	Provides identification and authentication for RSA	2048, 3072, 4096 bits - 112,	Public key certificate - PSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	cryptographic keys held in module	128, 152 bits				
EC-CERT	Provides identification and authentication for ECC cryptographic keys held in module	P-256, P-384 - 128, 192 bits	Public key certificate - PSP			
EC-KAS-PUB	Ephemeral EC public key used in key establishment (KAS) operation	P-256, P-384 - 128, 192 bits	Public key - PSP			KAS-2
EC-SM-CVC	Card Verifiable Certificate returned as part of Secure Messaging establishment	P-256, P-384 - 128 to 192 bits	Public key certificate - PSP	Counter DRBG		KAS-1

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
OS-DRBG-EI	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until power-cycle	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
OS-DRBG-SEED	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until power-cycle	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
OS-DRBG-KEY	Not Imported Not Exported	Stored in NVM :Plaintext		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
OS-DRBG-V	Not Imported Not Exported	Stored in NVM :Plaintext		Destroyed by termination of the module	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				(LifeCycle/Perform Zeroisation service)	
OS-SKEK	Entered during manufacturing/ personalization - plaintext storage	Stored in NVM :Plaintext		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
OS-MKEK	Not Imported Not Exported	Stored in NVM :Plaintext		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
SD-KENC	Exported using Approved KTS Encrypted using AES-CBC (using SD-KDEK) (RFC 3394 method) and transported using platform SCP03 Entered during manufacturing/ personalization	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	
SD-KMAC	Exported using Approved KTS Encrypted using AES-CBC (using SD-KDEK) (RFC 3394 method) and transported using platform	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	SCP03 Entered during manufacturing/ personalization				
SD-KDEK	Entered encrypted with the previous SD-KDEK Entered during manufacturing/ personalization	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	
SD-SENC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until power-cycle	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
SD-SMAC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until power-cycle	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
SD-RMAC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until power-cycle	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
DAP-DAPK	Entered during manufacturing/ personalization - plaintext storage	Stored in NVM :Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
RSA-PRIV-KEY	Imported in secure channel	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	specified by GP-Amd-I			Zeroisation service) Destroyed because of OS-MKEK zeroisation	
EC-PRIV-KEY	Imported in secure channel specified by GP-Amd-I	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	
EC-KAS-Z		Temporarily stored in RAM :Plaintext	Until session termination	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session	
AES-AUTH-INT	Imported in secure channel specified by GP-Amd-I	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
AES-AUTH-EXT	Imported in secure channel specified by GP-Amd-I	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
AUTH-PIN-L	Imported in secure channel specified by GP-Amd-I Entered in Plaintext	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
AUTH-PIN-G	Imported in secure channel specified by GP-Amd-I Entered in Plaintext	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AUTH-PC	Imported in secure channel specified by GP-Amd-I Entered in Plaintext	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
AUTH-PUK	Imported in secure channel specified by GP-Amd-I Entered in Plaintext	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
AUTH-PIN-HIST	Not Imported Not Exported	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
EC-SM-Z	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until derivation of EC-SM-SKENC, EC-SM-SKCMAC and EC-SM-SKRMAC	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared immediately following derivation of EC-SM-SKENC, EC-SM-SKCMAC and EC-SM-SKRMAC	
EC-SM-SKENC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until session termination	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session	
EC-SM-SKCMAC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until session termination	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				(deselect/reset/power off) or explicit closing of Secure Messaging session	
EC-SM-SKRMAC	Not Imported Not Exported	Temporarily stored in RAM :Plaintext	Until session termination	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session	
AES-AUTH-RND1	Exported in plaintext	Temporarily stored in RAM :Plaintext	Until session termination, receipt of client response or subsequent authentication command	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session Cleared upon receipt of client response Cleared on subsequent authentication command	
AES-AUTH-RND2	Entered in Plaintext	Temporarily stored in RAM :Plaintext	Until session termination, receipt of client response or subsequent authentication command	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session Cleared upon receipt of client response Cleared on	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				subsequent authentication command	
RSA-PUB-KEY	Imported in secure channel specified by GP-Amd-I Exported in plaintext	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	
EC-PUB-KEY	Imported in secure channel specified by GP-Amd-I Exported in plaintext	Stored in NVM - Encrypted:Encrypted	Until the OS-MKEK is zeroised	Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Destroyed because of OS-MKEK zeroisation	
RSA-CERT	Entered in Plaintext Exported in plaintext	Stored in NVM :Plaintext		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
EC-CERT	Entered in Plaintext Exported in plaintext	Stored in NVM :Plaintext		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	
EC-KAS-PUB	Not Exported Entered in Plaintext	Temporarily stored in RAM :Plaintext	Until session termination	Power-off Destroyed by termination of the module (LifeCycle/Perform Zeroisation service) Cleared on implicit (deselect/reset/power off) or explicit closing of Secure Messaging session	
EC-SM-CVC	Imported in secure channel specified by GP-Amd-I Exported in plaintext	Stored in NVM - Encrypted:Encrypted		Destroyed by termination of the module (LifeCycle/Perform Zeroisation service)	

Table 20: SSP Table 2

9.5 Transitions

- The usage of SHA-1 is deprecated through December 31, 2030, for non-digital signature applications and disallowed i.e., non-approved and not allowed thereafter. The module uses SHA-1 for non-digital signature applications, specifically, as the underlying hash function (PRF) for HMAC and NIST SP 800-56Cr1 HKDF.
- Usage of keys with a minimum-security strength of 112 bits i.e. RSA mod 2048 bits is deprecated through December 31, 2030, and disallowed thereafter. January 1, 2031, and beyond a minimum of 128 bits of security strength will be required.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware Integrity	32-bit CRC performed over all code located in Flash	32-bit CRC	SW/FW Integrity	9000	-

Table 21: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A2713) - Encrypt	128-bit	KAT	CAST	The APDU code 9000 signifies success	Encrypt	Performed automatically on every boot
AES-CBC (A2713) - Decrypt	128-bit	KAT	CAST	The APDU code 9000 signifies success	Decrypt	Performed automatically on every boot
AES-CMAC (A2713) - Encrypt	128-bit	KAT	CAST	The APDU code 9000	Encrypt	Performed automatically on every boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				signifies success		
AES-CMAC (A2713) - Decrypt	128-bit	KAT	CAST	The APDU code 9000 signifies success	Decrypt	Performed automatically on every boot
Counter DRBG (A2713)	256-bit	KAT	CAST	The APDU code 9000 signifies success	Health Tests: Generate, Reseed, Instantiate functions per Section 11 in NIST SP800-90Ar1	Performed automatically on every boot
ECDSA SigGen (FIPS186-5) (A6534)	P-521 SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Signature Generation	Performed automatically on every boot
ECDSA SigVer (FIPS186-5) (A6534)	P-521 SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Signature Verification	Performed automatically on every boot
HMAC-SHA2-256 (A2713)	SHA2-256	KAT	CAST	The APDU code 9000 signifies success	MAC Generation/Verification	Performed automatically on every boot
KAS-ECC-SSC Sp800-56Ar3 (A2713)	P-256 Scheme: onePassDh: KAS Role: initiator, responder	KAT	CAST	The APDU code 9000 signifies success	Shared Secret Computation (Z)	Performed automatically on every boot
KDA HKDF Sp800-56Cr1 (A2713)	SHA-1, SHA2-256, SHA2-384, SHA2-512	KAT	CAST	The APDU code 9000 signifies success	Key Derivation Function per NIST SP 800-56Cr1	Performed automatically on every boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF SP800-108 (A2713) - AES-128	Counter mode with AES-128	KAT	CAST	The APDU code 9000 signifies success	Key Derivation Function per NIST SP 800-108r1 (Counter mode)	Performed automatically on every boot
KDF SP800-108 (A2713) - HMAC-SHA-1	Feedback Mode with HMAC-SHA1	KAT	CAST	The APDU code 9000 signifies success	Key Derivation Function per NIST SP 800-108r1 (Feedback mode)	Performed automatically on every boot
RSA SigGen (FIPS186-5) (A6535)	2048-bit SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Signature Generation	Performed automatically on every boot
RSA SigVer (FIPS186-5) (A6535)	2048-bit SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Signature Verification	Performed automatically on every boot
SHA-1 (A2713)	SHA-1	KAT	CAST	The APDU code 9000 signifies success	Hash generation	Performed automatically on every boot
SHA2-256 (A2713)	SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Hash generation	Performed automatically on every boot
SHA2-512 (A2713)	SHA2-512	KAT	CAST	The APDU code 9000 signifies success	Hash generation	Performed automatically on every boot
Generate PCT	Pairwise consistency test performed when an	PCT	PCT	The APDU code 9000	-	Performed automatically on generation of keypairs

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	asymmetric key pair is generated for RSA or ECC.			signifies success		
Signature PCT	Pairwise consistency test performed when a signature is generated for RSA or ECDSA	PCT	PCT	The APDU code 9000 signifies success	-	Performed automatically on generation of keypairs
Firmware Load Test	ECDSA P-256 with SHA2-256	Load Test	SW/FW Load	The APDU code 9000 signifies success	Signature Verification based on ECDSA P-256 with SHA2-256	Upon loading of firmware from an external source (e.g. the OpenFIPS201 applet)
NIST SP800-90B ENT (P) Repetition Count Test (RCT)	-	NIST SP 800-90B Health Test	CAST	Implicit, based on output of entropy	-	Performed continuously by the entropy source
NIST SP800-90B ENT (P) Developer Defined Heath Test Transition Count Test	-	NIST SP 800-90B Health Test	CAST	Implicit, based on output of entropy	-	Performed continuously by the entropy source
NIST SP800-90B ENT (P) Developer Defined	-	NIST SP 800-90B Health Test	CAST	Implicit, based on output of entropy	-	Performed continuously by the entropy source

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Heath Test ChiSquare Test						
NIST SP800-90B ENT (P) Developer Defined Heath Test Amplitude Limiter Analog Test	-	NIST SP 800-90B Health Test	CAST	Implicit, based on output of entropy	-	Performed continuously by the entropy source
OneStep KDF	SHA2-256	KAT	CAST	The APDU code 9000 signifies success	Key Derivation Function per NIST SP 800-56Cr1	Performed automatically prior to first use

Table 22: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware Integrity	32-bit CRC	SW/FW Integrity	Repeated after every 500,000 CAPDUs/commands	Automatic execution per module design (the test is repeated after every 500,000 CAPDUs/commands)

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A2713) - Encrypt	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
AES-CBC (A2713) - Decrypt	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
AES-CMAC (A2713) - Encrypt	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
AES-CMAC (A2713) - Decrypt	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
Counter DRBG (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
ECDSA SigGen (FIPS186-5) (A6534)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
ECDSA SigVer (FIPS186-5) (A6534)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
HMAC-SHA2-256 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
KAS-ECC-SSC Sp800-56Ar3 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
KDA HKDF Sp800-56Cr1 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
KDF SP800-108 (A2713) - AES-128	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
KDF SP800-108 (A2713) - HMAC-SHA-1	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
RSA SigGen (FIPS186-5) (A6535)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
RSA SigVer (FIPS186-5) (A6535)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
SHA-1 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
SHA2-256 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
SHA2-512 (A2713)	KAT	CAST	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
Generate PCT	PCT	PCT	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
Signature PCT	PCT	PCT	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
Firmware Load Test	Load Test	SW/FW Load	On Demand	Manually, all the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00
NIST SP800-90B ENT (P) Repetition Count Test (RCT)	NIST SP 800-90B Health Test	CAST	On Demand	Manually, by rebooting the module/resetting the entropy source
NIST SP800-90B ENT (P) Developer Defined Heath Test Transition Count Test	NIST SP 800-90B Health Test	CAST	On Demand	Manually, by rebooting the module/resetting the entropy source
NIST SP800-90B ENT (P) Developer Defined Heath Test ChiSquare Test	NIST SP 800-90B Health Test	CAST	On Demand	Manually, by rebooting the module/resetting the entropy source
NIST SP800-90B ENT (P) Developer Defined Heath Test Amplitude Limiter Analog Test	NIST SP 800-90B Health Test	CAST	On Demand	Manually, by rebooting the module/resetting the entropy source

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
OneStep KDF	KAT	CAST	On Demand	Manually, by initiating a PIV Secure Messaging session

Table 24: Conditional Periodic Information

The JCOP OS is intended to execute pre-operational self-tests (rather than conditional self-tests) periodically, since these tests are pre-defined to be executed post resets. The periodic time will thus always start from zero. RAM can be used to manage the periodic time interval which is the number of execution events e.g., number of APDUs/commands received (the tests are repeated after every 500,000 CAPDUs/commands).

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard Error (MUTE) state	All the Self-Tests can be performed on-demand with the GET DATA APDU command (Info service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00; The expected result is FE04DF4B0120; The return code 9000 signifies success; In case of a failure, the module enters a hard error (MUTE) state, returns a code/status indicator and inhibits all cryptographic functions	In case of failure of a pre-operational, conditional self-test or NIST SP 800-90B compliant entropy source health test	A reset of the module can be attempted but in the event that the error persists, the module must be returned to the vendor	66A7

Table 25: Error States

10.5 Operator Initiation of Self-Tests

The pre-operational self-test must be completed successfully prior to any other use of cryptography by the module. The Cryptographic Algorithm Self-Tests are either performed at boot or prior to first use. The conditional self-tests are performed when the corresponding conditions occur. If one of the self-tests fails, the system is halted and will start again after a reset.

All the Self-Tests can be performed on-demand with the GP GET DATA APDU command (*Info* service) with the following parameters: CLA = 80, INS = CA, P1 = 00, P2 = FE, Lc = 04, Incoming Data = DF4B0120, and Le = 00. The expected result is FE04DF4B0120.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

All configuration management items are managed using an automated configuration management system. The module is designed to allow the testing of all provided security-related services. All firmware is implemented using a high-level language and is designed in a manner that avoids the use of code, parameters, or symbols not necessary for the module's functionality and execution.

The module is delivered in the unconfigured state, i.e., non-compliant state. To comply with and maintain the FIPS 140-3 validation, it would be the CO's responsibility to enable the Approved mode of operation as follows:

1. Acquire genuine, factory condition NXP P71D600 tokens from an authorised source.
2. Initialise the token according to the desired configuration, paying attention to the JCOP Platform Requirements section below.
3. Download, load and install a FIPS 140-3 validated OpenFIPS201 Applet binary.
4. Initialise and pre-personalise the Applet as required by your system, paying attention to the Security Rules section below.
5. Personalise the Applet according to the NIST PIV Specification

The essential requirements to deploy OpenFIPS201 in the Approved mode are:

1. You must use the NXP P71D600 smart card platform. This is currently the only hardware platform certified for FIPS 140-3.
   NOTE: This rule also applies to any other FIPS 140-3 validated smart card platform.
2. The hardware platform must be configured and deployed according to the User Guidance and Administration manual provided, with care to ensure none of its Approved mode restrictions are bypassed.
3. You must use the OpenFIPS201 FIPS binary only, as provided and signed on the Release page.
   NOTE: Even re-compiling the exact source code of the FIPS validated version will cause your certificate to be considered invalid and should be done at your own risk.
4. The configuration and pre-personalisation of your applet must adhere to the rules described below.
5. No other applet may be installed on the card.

11.2 Administrator Guidance

The aforementioned information can also be found in the *JCOP 4.5 User guidance and administrator manual and the OpenFIPS201 User Guidance and Administration* documents.

11.3 Non-Administrator Guidance

The aforementioned information can also be found in the *JCOP 4.5 User guidance and administrator manual and the OpenFIPS201 User Guidance and Administration* documents.

11.4 Design and Rules

JCOP Platform Requirements

- The hardware platform used must be the one associated with this FIPS 140-3 certificate.
- The specific configuration of the above hardware configuration when ordering from NXP or an approved reseller is obtained by specifying the Order Entry Form (OEF) identifier, which for the Module is BA43.
- The card life-cycle state shall be set to SECURED prior to operational use.
- GlobalPlatform SCP01 shall not be enabled or used by the card platform for any function.
- GlobalPlatform SCP02 shall not be enabled or used by the card platform for any function.
- If applicable, the card shall be transitioned from its transport state prior to use.
- The ISD shall be configured with customer-specific ISD SCP03 key values, generated using an approved RBG.
    - ISD Transport Key values shall not be used.
    - Known default and test ISD keys shall not be used.
- The FIPS 140-3 validated Applet package shall be loaded and installed under either the Issuer Security Domain (ISD) or a Supplementary Security Domain (SSD) created specifically for the Applet.
- The Approved OpenFIPS201 binary is signed with a ‘Data Access Pattern’ (DAP) digital signature to provide authentication and integrity as per the GlobalPlatform specification. The following must be true for the Operating Platform prior to Applet load:
    - The applicable DAP public key (see Applet release notes) must be configured on the applicable security domain.
    - The Mandated DAP privilege must be set on the applicable Security Domain.
- No other Applet or Package shall be installed or configured onto the target Platform.
- The config module shall be deleted prior to operational use.
- The firmware is designed and implemented in a manner that avoids the use of code, parameters or symbols not necessary for the module's functionality and execution.

2. OpenFIPS201 Applet Requirements

- The Applet binary must be selected from the FIPS validated list of OpenFIPS201 Applet releases.

- The Applet must be loaded under DAP as per the above Operating Platform requirements.

- The use of SCP-03 is mandatory for all administrative functions except for the PIV administration commands (PUT DATA and GENERATE ASYMMETRIC KEYPAIR) , which may use a 9B key configured with the following settings:
    - The key role Authenticate must be enabled
    - The key attribute Permit Internal must not be set.
    - The key attribute Permit External must not be set.
    - The key attribute Importable must be set.

- It is strongly advised to never use the 9B key for any kind of administration, as it does not provide encryption, integrity or replay protection of any kind.

- The following limitations are enforced by the Applet in the Approved mode:
    - The TDEA192 mechanism is not permitted and any attempt to create a key of this type will fail.
    - The RSA-1024 mechanism is not permitted and any attempt to create a key of this type will fail.
    - ECDSA operations on SHA1 hash values will not be permitted and an error status will return.
    - ECDSA operations on SHA-256 hash values will not be permitted for ECC-P384 key types and an error status will return.
    - PIN objects will not permit usage on the contactless interface without either SCP03 or a PIV Secure Messaging established.
    - At least one User authentication method must be enabled, being either of the Local PIN for the Global PIN for the User role to be permitted.
    - For any active PIN, the minimum length attribute is limited by Applet to a value of 6 or higher in the Approved mode.
    - For the PUK, the minimum length attribute is limited by Applet to a value of 6 or higher in the Approved mode.

11.5 Maintenance Requirements

No specific maintenance requirements apply to the module.

11.6 End of Life

The module must be zeroised (using the Perform zeroisation service per Section 4 of this document) in order to perform secure sanitization of the module.

12 Mitigation of Other Attacks

12.1 Attack List

The module is protected against the following non-invasive attacks: SPA, DPA, Timing Analysis and Fault Induction using a combination of firmware and hardware countermeasures. Protection

features include detection of out-of-range supply voltages, frequencies or temperatures, fault induction mitigations like light sensors, voltage glitch sensors and an active shield, and detection of illegal addresses or instruction.

12.2 Mitigation Effectiveness

All cryptographic computations and sensitive operations such as critical data comparison provided by the module are designed to be resistant to timing and power analysis. Sensitive operations are performed in constant time, regardless of the execution context (parameters, keys, etc.), owing to a combination of hardware and firmware features. In addition to the non-invasive attacks, the module also uses standard passivation techniques and is protected by active shielding (a grid of top metal layer wires with tamper response) which qualifies for classification under mitigation of other attacks.