eToken 5110

FIPS 140-2 Cryptographic Module
Non-Proprietary Security Policy Level 3

Table of Contents

References 5
Acronyms and definitions 6
1 Introduction 7
1.1 eToken Applet 8
1.2 eTPnP applet is associated to eToken applet and offers: 8
2 Hardware and Physical Cryptographic Boundary 8
2.1 eToken 5110 Crypto Boundary 9
2.2 Ports and Interfaces 9
3 Cryptographic Module Specification 10
3.1 USB MCU Firmware and Logical Cryptographic Boundary 10
3.2 SC OS Firmware and Logical Cryptographic Boundary 11
3.3 USB MCU FW Versions and mode of operation 12
3.3.1 Get Descriptors (VSR 0xA0) Device to Host 12
3.4 SC Versions and mode of operation 14
3.5 Cryptographic Functionality 17
4 Platform Critical Security Parameters 18
4.1 eToken Applet Critical Security Parameters 20
4.2 USB MCU FW Critical Security Parameters 21
5 Roles, Authentication and Services 21
5.1 Secure Channel Protocol (SCP) Authentication 22
5.2 eToken Applet Authentication 22
5.3 FW Updater Authentication 23
5.4 Platform Services 23
5.5 eToken Applet Services 25
5.6 USB MCU FW Services 27
5.7 eTPnP Applet Services 28
6 Finite State Model 28
7 Physical Security Policy 28
8 Operational Environment 28
9 Electromagnetic Interference and Compatibility (EMI/EMC) 28
10 Self-test 29
10.1 Power-on Self-test 29
10.2 Conditional Self-tests 30
11 Design Assurance 30
11.1 Configuration Management 30
11.2 Delivery and Operation 30

11.3 Guidance Documents 30
11.4 Language Level 30
12 Mitigation of Other Attacks Policy 31
13 Security Rules and Guidance 31

Table of Tables

Table 1 – References 6
Table 2 – Acronyms and Definitions 6
Table 3 – Security Level of Security Requirements 7
Table 4 – USB Physical Interfaces 9
Table 5 – USB Logical Interfaces 10
Table 6 - Get Descriptors (VSR 0xA0) Device to Host 12
Table 7 - Get Descriptors - Header 13
Table 8 – Kernel Info (Index 0) 13
Table 9 - GET DATA Command 14
Table 10 - Versions and Mode of Operations Indicators (tag 9F-7F) 15
Table 11 – Versions and Mode of Operations Indicators (tag 0103) 16
Table 12 – Get Data Applet Version 16
Table 13 – eToken Version Returned Values 16
Table 14 – FIPS Approved Cryptographic Functions 18
Table 15 – Non-FIPS Approved But Allowed Cryptographic Functions 18
Table 16 - Platform Critical Security Parameters 19
Table 17 – eToken Applet Critical Security Parameters 20
Table 18 – USB MCU FW Critical Security Parameters 21
Table 19 - Role Description 21
Table 20 - Unauthenticated Services and CSP Usage 23
Table 21 – Authenticated Card Manager Services and CSP Usage 24
Table 22 – eToken Applet Services and CSP Usage 27
Table 23 – USB MCU FW applet Services 27
Table 24 – eTPnP applet Services 28
Table 25 – Power-On Self-Test 29

Table of Figures

Figure 1 – eToken 5110 Crypto Boundary (Top and Bottom) 9
Figure 2 – Shows the Module with the outer enclosure, which is not within the cryptographic boundary 9
Figure 3 – USB MCU Block Diagram 10
Figure 4 – SC Module Block Diagram 11

References

Acronym	Full Specification Name
[FIPS140-2]	NIST, *Security Requirements for Cryptographic Modules*, May 25, 2001
[GlobalPlatform]	*GlobalPlatform Consortium: GlobalPlatform Card Specification 2.1.1,* March 2003, http://www.globalplatform.org *GlobalPlatform Consortium: GlobalPlatform Card Specification 2.1.1* Amendment A, March 2004 *GlobalPlatform Consortium: GlobalPlatform Card Specification 2.2* Amendment D, Sept 2009
[ISO 7816]	ISO/IEC 7816-1:2003 *Identification cards -- Integrated circuit(s) cards with contacts -- Part 1: Physical characteristics* ISO/IEC 7816-2:2007 *Identification cards -- Integrated circuit cards -- Part 2: Cards with contacts -- Dimensions and location of the contacts* ISO/IEC 7816-3:2006 *Identification cards -- Integrated circuit cards -- Part 3: Cards with contacts -- Electrical interface and transmission protocols* ISO/IEC 7816-4:2013 *Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange*
[ISO 14443]	*Identification cards – Contactless integrated circuit cards – Proximity cards* ISO/IEC 14443-1:2008 Part 1*: Physical characteristics* ISO/IEC 14443-2:2010 Part 2: *Radio frequency power and signal interface* ISO/IEC 14443-3:2011 Part 3: *Initialization and anticollision* ISO/IEC 14443-4:2008 Part 4: *Transmission protocol*
[JavaCard]	*Java Card 2.2.2 Runtime Environment (JCRE) Specification* *Java Card 2.2.2 Virtual Machine (JCVM) Specification* *Java Card 2.2.2 Application Programming Interface* *Java Card 3.0.5 Application Programming Interface [only for ECDSA with SHA2, AES-CMAC]* Published by Sun Microsystems
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, Revision 1, November 2015
[SP 800-90A]	NIST Special Publication 800-90, *Recommendation for the Random Number Generation Using Deterministic Random Bit Generators,* - revision 1, June 2015.
[SP 800-108]	NIST Special Publication 800-108, *Recommendation for Recommendation for Key Derivation Using Pseudorandom Functions,* October 2009.
[SP 800-67]	NIST Special Publication 800-67, *Recommendation for the Triple Data Encryption Algorithm (Triple-DES) Block Cipher*, - revision 1, January 2012.
[FIPS 113]	NIST, *Computer Data Authentication*, FIPS Publication 113, 30 May 1985.
[FIPS 197]	NIST, *Advanced Encryption Standard (AES)*, FIPS Publication 197, November 26, 2001.
[PKCS#1]	*PKCS #1 v2.2: RSA Cryptography Standard*, RSA Laboratories, October 27, 2012.
[FIPS 186-4]	NIST, Digital Signature Standard (DSS), FIPS Publication 186-4, July, 2013

Acronym	Full Specification Name
[SP 800-56A]	NIST Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, revision 2, May 2013.
[FIPS 180-3]	NIST, *Secure Hash Standard*, FIPS Publication 180-3, October 2008
[AESKeyWrap]	NIST, *AES Key Wrap Specification*, 16 November 2001. This document defines symmetric key wrapping, Use of 2-Key Triple-DES in lieu of AES is described in [IG] D.2.
[IG]	NIST, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program,* last updated 10 May 2016.
[USB 2.0]	USB.org, Universal Serial Bus Revision 2.0 specification

Table 1 – References

Acronyms and definitions

Acronym	Definition
GP	Global Platform
CVC	Card Verifiable Certificate
MMU	Memory Management Unit
OP	Open Platform
RMI	Remote Method Invocation
SC	Secure Controller

Table 2 – Acronyms and Definitions

1 Introduction

This document defines the Security Policy for the Gemalto SafeNet eToken 5110 which comprises the 5110 USB MCU FW, the IDCore 30-revB platform and the eToken Applet 1.8 and herein denoted as Cryptographic Module. The Cryptographic Module or CM, validated to FIPS 140-2 overall Level 3, is a USB token that contains a secure controller (SC) module implementing the Global Platform operational environment, with Card Manager, the eToken Applet 1.8.

The CM is a limited operational environment under the FIPS 140-2 definitions. The CM SC includes a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires a separate FIPS 140-2 validation. The CM also includes the USB MCU FW firmware load service to support necessary updates of the USB controller FW.

The FIPS 140-2 security levels for the Module are as follows:

Security Requirement	Security Level
Cryptographic Module Specification	3
Cryptographic Module Ports and Interfaces	3
Roles, Services, and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

Table 3 – Security Level of Security Requirements

The CM implementation is compliant with:

- [ISO 7816] Parts 1-4
- [JavaCard]
- [GlobalPlatform]
- [USB 2.0]

1.1 eToken Applet

eToken Applet (v1.8) is a Java applet that provides all the necessary functions to integrate a smart card in a public key infrastructure (PKI) system, suitable for identity and corporate security applications. It is also useful for storing information about the cardholder and any sensitive data. eToken Applet implements state–of–the–art security and conforms to the latest standards for smart cards and PKI applications. It is also fully compliant with digital signature law.

The eToken is designed for use on JavaCard 2.2.2 and Global Platform 2.1.1 compliant smart cards.

The main features of eToken Applet are as follows:

- Digital signatures—these are used to ensure the integrity and authenticity of a message. (RSA, ECDSA)
- Storage of sensitive data based on security attributes
- Secure messaging based on the Triple-DES 3 Keys algorithms.
- Public key cryptography, allowing for RSA keys and ECDSA keys
- Storage of digital certificates—these are issued by a trusted body known as a certification authority (CA) and are typically used in PKI authentication schemes.
- Decryption RSA , ECDH
- On board key generation (RSA, ECDSA)
- Support of integrity on data to be signed based on the Secure messaging protocol.

1.2 eTPnP applet is associated to eToken applet and offers:

- GUID tag reading, defined in Microsoft Mini Driver specification.

2 Hardware and Physical Cryptographic Boundary

eToken 5110 is a multiple-Chip standalone cryptographic module. Two (2) ICs are mounted on a PCB assembly with a connector and passive components, covered by epoxy on both sides, exposing only the LED and USB connector. The Module is intended to be covered within a plastic enclosure. Physical inspection inside the Module boundary is not practical, as the epoxy layer is opaque.

The Module meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations.

2.1 eToken 5110 Crypto Boundary

Figure 1 – eToken 5110 Crypto Boundary (Top and Bottom)

Figure 2 – Shows the Module with the outer enclosure, which is not within the cryptographic boundary

2.2 Ports and Interfaces

The Module functions as a slave device to process and respond to commands.

This module provides a contact interface that is fully compliant with USB 2.0.

Interface	Description
USBDM	USB D- differential data
USBDP	USB D+ differential data
V_{Bus}	Power supply input
GND	Ground (reference voltage)
LED	LED indicator

Table 4 – USB Physical Interfaces

The I/O ports of the platform provide the following logical interfaces:

Interface	USB
Data In	USBDM, USBDP
Data Out	USBDM, USBDP
Status Out	USBDM, USBDP, LED
Control In	USBDM, USBDP

Table 5 – USB Logical Interfaces

3 Cryptographic Module Specification

3.1 USB MCU Firmware and Logical Cryptographic Boundary

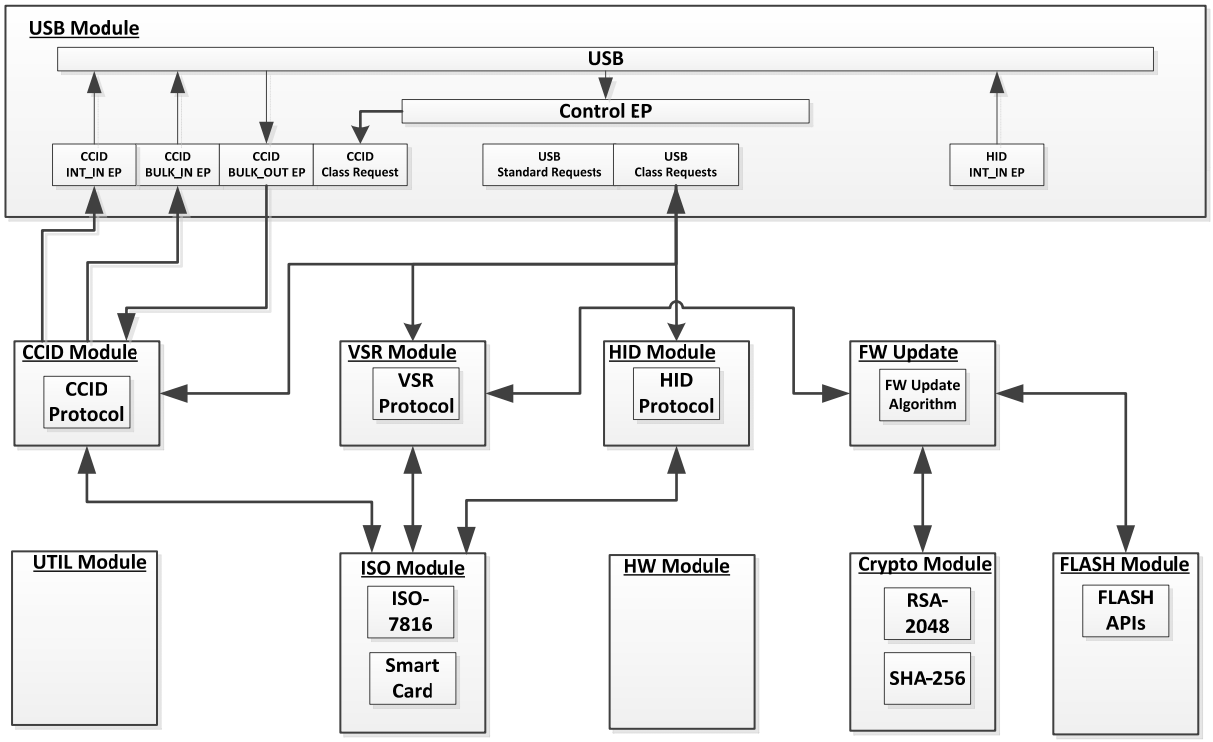

Figure 3 – USB MCU Block Diagram

The CM provides framework for the USB Standard and Class requests including the API dedicated to the CCID protocol, VSR propriety protocol, and the HID propriety protocol. The CM defines an interface for USB MCU firmware update service secured with RSA-2048 PKCS#1 RSASSA-PKCS1-v1_5 signature. The USB MCU FW communicates with the SC OS using ISO-7816 T1 protocol.

3.2 SC OS Firmware and Logical Cryptographic Boundary

Figure 4 below depicts the Module operational environment and applets.

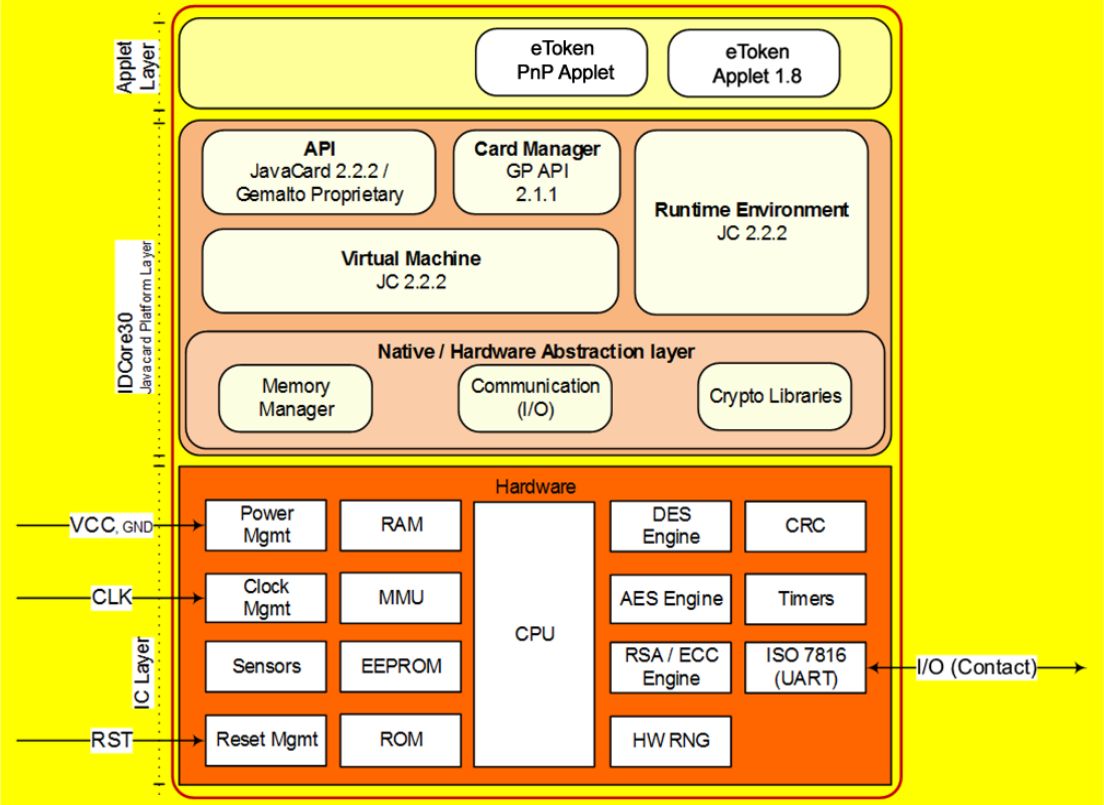

Figure 4 – SC Module Block Diagram

The CM supports [ISO7816] T=1 communication protocols.

The CM provides services to both external devices and internal applets as the eToken Applet 1.8.

Applets, as eToken Applet 1.8, access module functionalities via internal API entry points that are not exposed to external entities. External devices have access to CM services by sending APDU commands.

The CM provides an execution sandbox for the eToken Applet and performs the requested services according to its roles and services security policy.

The CM inhibits all data output via the data output interface while the module is in error state and during self-tests.

The *JavaCard API* is an internal interface, available to applets. Only applet services are available at the card edge (the interfaces that cross the cryptographic boundary).

The *Javacard Runtime Environment* implements the dispatcher, registry, loader, logical channel and RMI functionalities.

The *Virtual Machine* implements the byte code interpreter, firewall, exception management and byte code optimizer functionalities.

The *Card Manager* is the card administration entity – allowing authorized users to manage the card content, keys, and life cycle states.

The *Memory Manager* implements services such as memory access, allocation, deletion, garbage collector.

The *Communication* handler deals with the implementation of ATR, PSS, and T=1 protocols.

The *Cryptography Libraries* implement the algorithms listed in Section 3.5.

3.3 USB MCU FW Versions and mode of operation

Hardware: STM32F042K6U6TR

Firmware: 5110 FIPS FW ver-15.0

The MCU FW version is retrieved via VSR 0xA0; The FW version is encoded in the

kernelVerMajor, **kernelVerMinor**, **kernelBuild** Fields.

3.3.1 Get Descriptors (VSR 0xA0) Device to Host

- This command is being implemented in the **KERNEL** for Kernel Info
- One Stage command – Only CTRL Read.
- Each Descriptor starts with a fixed Header, and might have a string descriptor.

Field	Offset	Length	Value	Description
Request type	0	1	0xC0	Device to Host VSR
Request	1	1	0xA0	Get Descriptor
Value	2	2		0x0 – For Kernel Info 0x6 – For String descriptor (not implemented)
Index	4	2	Index	If wValue == 6 then it's the Index of requested string descriptor: • VENDOR 0x01 • PRODUCT 0x02
Data Length	6	2	According to the descriptor	

Table 6 - Get Descriptors (VSR 0xA0) Device to Host

3.3.1.1 Get Descriptors - Header

Get Descriptors Header has a fixed structure with fixed size common to all Descriptors.

Offset	Length	Value	Field Name	Description
0	2	N	size	Descriptor size including the header
2	1	1	version	Descriptor Version set to 1
3	1	0	stringIndex	Index of string descriptor, 0 value means that there is no string descriptor

Table 7 - Get Descriptors – Header

3.3.1.2 Kernel Info (Index 0)

Offset	Length	Value	Field Name	Description
0	4		descriptorHeader	Descriptor Header
4	1		kernelLocation	0x0 - Low, 0x1 - High
5	2	2	kernelType	0x0 – NG PRO T1 CARDOS 0x1 - PRO T0 JAVA 0x2 - NG PRO T1 JAVA
7	2	15	kernelVerMajor	Kernel Version – firmware major version, changing the protocol will affect the FW major version.
9	2	0	kernelVerMinor	Kernel Version – Firmware minor version
11	4	X	kernelBuild	Firmware build
15	2	1	microController	Micro Controller type: 0 - Micro Controller C8051F320/1 with 16K EE 1 - Micro Controller C8051F387 2 - Microcontrollers ST STM32F042
17	4		tokenId	Token Id
21	2		tokenIdEx	Token Id Extender
23	1	0	status	status

Table 8 – Kernel Info (Index 0)

3.4 SC Versions and mode of operation

Hardware: SLE78CFX3000PH
Firmware: IDCore30-revB - Build 06, eToken Applet version 1.8 and eTPnP Applet V1.0

The CM is always in the approved mode of operation. To verify that a CM is in the approved mode of operation, select the Card Manager and send the GET DATA commands shown below:

Field	CLA	INS	P1-P2 (Tag)	Le (Expected response length)	Purpose
Value	00	CA	9F-7F	2A	Get CPLC data
			01-03	1D	Identification information (proprietary tag)

Table 9 - GET DATA Command

The CM responds with the following information:

IDC30-revB - CPLC data (tag 9F7F)			
Byte	**Description**	**Value**	**Value meaning**
1-2	IC fabricator	**4090h**	Infineon
3-4	IC type	**7901**	SLE78CFX3000PH
5-6	Operating system identifier	**1291**	Gemalto
7-8	Operating system release date (YDDD) – Y=Year, DDD=Day in the year	**5356**	Operating System release Date
9-10	Operating system release level	**0200h**	V2.0
11-12	IC fabrication date	xxxxh	Filled in during IC manufacturing
13-16	IC serial number	xxxxxxxxh	Filled in during IC manufacturing
17-18	IC batch identifier	xxxxh	Filled in during IC manufacturing
19-20	IC module fabricator	xxxxh	Filled in during module manufacturing
21-22	IC module packaging date	xxxxh	Filled in during module manufacturing
23-24	ICC manufacturer	xxxxh	Filled in during module embedding
25-26	IC embedding date	xxxxh	Filled in during module embedding
27-28	IC pre-personalizer	xxxxh	Filled in during smartcard preperso

29-30	IC pre-personalization date	xxxxh	Filled in during smartcard preperso
31-34	IC pre-personalization equipment identifier	xxxxxxxxh	Filled in during smartcard preperso
35-36	IC personalizer	xxxxh	Filled in during smartcard personalization
37-38	IC personalization date	xxxxh	Filled in during smartcard personalization
39-42	IC personalization equipment identifier	xxxxxxxxh	Filled in during smartcard personalization

Table 10 - Versions and Mode of Operations Indicators (tag 9F7F)

IDC30-revB - Identification data (tag 0103)			
Byte	**Description**	**Value**	**Value meaning**
1	Gemalto Family Name	**B0**	Javacard
2	Gemalto OS Name	**84**	IDCore family (OA)
3	Gemalto Mask Number	**56**	G286
4	Gemalto Product Name	**51**	IDCore30-revB
5	Gemalto Flow Version	**XY**	**X** is the type of SCP: ▪ 2xh for SCP0300 flows ▪ 3xh for SCP0310 flows **Y**: is the version of the flow (x=1 for version 01). For instance: ▪ **21h** = SCP0300 - flow 01 (version 01) ▪ **31h** = SCP0310 - flow 01 (version 01)
6	Gemalto Filter Set	**00**	▪ Major nibble: filter family = 00h ▪ Lower nibble: version of the filter = 00h
7-8	Chip Manufacturer	**4090**	Infineon
9-10	Chip Version	**7901**	SLE78CFX3000PH
11-12	FIPS configuration	**8D00**	MSByte: b8 : 1 = **conformity to FIPS certificate** b7 : 0 = not applicable b6 : 0 = not applicable b5 : 0 = not applicable

			b4 : 1 = ECC supported b3 : 1 = RSA CRT supported b2 : 1 = RSA STD supported b1 : 1 = AES supported LSByte: b8 .. b5 : 0 = not applicable b4 : 0 = not applicable (ECC in contactless) b3 : 0 = not applicable (RSA CRT in contactless) b2 : 0 = not applicable (RSA STD in contactless) b1 : 0 = not applicable (AES in contactless) For instance: **8F 00** = FIPS enable (CT only)–AES-RSA CRT/STD-ECC **(Full FIPS)** **8D 00** = FIPS enable (CT only)–AES-RSA CRT-ECC **(FIPS PK CRT) *** **85 00** = FIPS enable (CT only)–AES-RSA CRT **(FIPS RSA CRT)** **00 00** = FIPS disable (CT only)–No FIPS mode **(No FIPS)** **(* default configuration)**
13	FIPS Level for IDPrime MD product	**00**	03 = FIPS Level 3
14-29	RFU	**xx..xxh**	-

Table 11 – Versions and Mode of Operations Indicators (tag 0103)

The **eToken Applet 1.8** is identified with an applet version

Field	CLA	INS	P1-P2 (Tag)	Le (Expected response length)	Purpose
Value	**00**	**CA**	**01-11**	**06**	Get Data Applet Version

Table 12 – Get Data Applet Version

The **eToken Applet 1.8** Version is returned in TLV format as follows:

Tag	Length	Value	Value meaning
11	04	1.8.XX	VerMajor (one byte), VerMinor (one byte), Build (two bytes);

Table 13 – eToken Version Returned Values

3.5 Cryptographic Functionality

The Module operating system implements the FIPS Approved and Non-FIPS Approved cryptographic function listed in Tables below.

Algorithm	Description	Cert #
AES	[FIPS 197] Advanced Encryption Standard algorithm. The Module supports 128, 192, and 256-bit key lengths with ECB and CBC modes.	3779
AES CMAC	AES CMAC The Module supports 128-, 192- and 256-bit key lengths.	3779
CVL (ECC-CDH)	[SP 800-56A] The Section 5.7.1.2 ECC CDH Primitive. The Module is CAVP validated for the NIST defined P-224, P-256, P-384 and P-521 curves.	719
CVL (RSASP1)	[FIPS 186-4] RSA PKCS1-v1.5 signature generation primitive	803
CVL (RSADP)	[SP 800-56B] RSA PKCS#1 v2.1 decryption operation primitive component as specified in Section 7.1.2.Section 5.1.2 decryption primitive	804
DRBG	[SP 800-90] Deterministic Random Number Generators [CTR_DRBG mode based on AES]	1045
ECDSA	[FIPS 186-4] Elliptic Curve Digital Signature Algorithm: signature generation, verification and key pair generation. The Module is CAVP validated for the NIST defined P-224, P-256, P-384 and P-521 curves. (Note: Sig Ver P-192 is not used by the module.)	814
KBKDF	[SP 800-108] KDF for AES CMAC. The Module supports 128-, 192- and 256-bit key lengths.	81
KTS	[SP800-38F] Symmetric Key wrapping using 128, 192, or 256 bit keys (based on AES and AES CMAC Cert. #3779), meets the SP800-38F §3.1 ¶3. Key establishment methodology provides 128, 192, or 256 bits of strength.	3779
KTS	[SP800-38F] Symmetric Key wrapping using 3-key Triple-DES Cert. #2100 and Triple-DES MAC Vendor Affirmed), meets the SP800-38F §3.1 ¶3. Key establishment methodology provides 112 bits of strength.	2100
RSA	[FIPS 186-4] RSA signature generation, verification, and key pair generation. The Module follows PKCS#1 and is CAVP validated for 2048 bit key length. (Note: Sig Ver 1024 is not used by the module.)	1946
RSA CRT	[FIPS 186-4] RSA signature generation, verification, CRT key pair generation. The Module follows PKCS#1 and is CAVP validated for 2048 bit key length.	1947
RSA Signature Verification	[FIPS 186-4] USB FW implementations of RSA Signature Verification	2037
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	[FIPS 180-4] Secure Hash Standard compliant one-way (hash) algorithms.	3146
Triple-DES	[SP 800-67] Triple Data Encryption Algorithm. The Module supports the 3-Key options; CBC and ECB modes. Note that the Module does not support a mechanism that would allow collection of plaintext / ciphertext pairs aside from authentication, limited in use by a counter.	2100

Triple -DES MAC	[FIPS 113] Triple-DES Message Authentication Code. Vendor affirmed, based on validated Triple-DES.	Vendor Affirmed
SHA 256	[FIPS 180-4] USB FW implementations of Secure Hash Standard compliant one-way (hash) algorithms	3276

Table 14 – FIPS Approved Cryptographic Functions

Algorithm	Description
EC Diffie-Hellman key agreement	SP 800-56A; non-compliant - key agreement using NIST defined, P-224, P-256, P-384 and P-521 curves. Key establishment methodology provides 112, 128, or 192 bits of strength.
NDRNG	Used to initialize the CTR DRBG. Provides more than 112 bits of entropy.

Table 15 – Non-FIPS Approved But Allowed Cryptographic Functions

The CM includes an uncallable DES implementation. This algorithm is not used and no security claims are made for its presence in the Module.

FIPS approved security functions used specifically by the **eToken Applet** are:

- **DRBG**
- **AES**
- **RSA**
- **ECDSA**
- **SHA-256,**
- **ECDH**
- **Triple-DES 3 Key**

(Note: no security function is used in **eTPnP applet**)

4 Platform Critical Security Parameters

All CSPs used by the CM are described in this section. All usages of these CSPs by the CM are described in the services detailed in Section 5.

Key	Description / Usage
OS-RNG-SEED-KEY	256-bit random drawn by the NDRNG HW chip (AIS-31PTG.2), used as a seed key for the [SP 800-90A] DRBG implementation.
OS-RNG-STATE	16-byte random value and 16-byte counter value used in the [SP 800-90] DRBG implementation. 16-byte AES state V and 16-byte AES key used in the [SP800-90A] CTR DRBG implementation.
OS-GLOBALPIN	6 to 16 bytes Global PIN value managed by the ISD. Character space is not restricted by the module.

OS-MKDK	AES-128/192/256 (SCP03) key used to encrypt OS-GLOBALPIN value
SD-KENC	AES-128/192/256 (SCP03) Master key used by the CO role to generate SD-SENC
SD-KMAC	AES-128/192/256 CMAC (SCP03) Master key used by the CO role operator to generate SD-SMAC
SD-KDEK	AES-128/192/256 (SCP03) Sensitive data decryption key used by the CH role to decrypt CSPs for SCP03.
SD-SENC	AES-128/192/256 (SCP03) Session encryption key used by the CO role to encrypt / decrypt secure channel data.
SD-SMAC	AES-128/192/256 CMAC (SCP03) Session MAC key used by the CO role to verify inbound secure channel data integrity.
SD-SDEK	AES-128/192/256 (SCP03) Session DEK key used by the CO role to decrypt CSPs.
DAP-SYM	AES-128/192/256 (SCP03) key optionally loaded in the field and used to verify the signature of packages loaded into the Module.

Table 16 - Platform Critical Security Parameters

Keys with the "SD-" prefix pertain to a Global Platform Security Domain key set. The module supports the Issuer Security Domain at minimum, and can be configured to support Supplemental Security Domains.

4.1 eToken Applet Critical Security Parameters

Key	Description / Usage
ID_AUTH_OP	Triple-DES 3 Key MAC key used to authenticate the ESO or CH role using a challenge-response protocol.
ID_SM_ENC_IN_OP	Triple-DES 3 SM Decryption key used to decrypt data sent by the host application as a part of the operator operations.
ID_SM_MAC_IN_OP	Triple-DES 3 SM MAC key used to guarantee integrity on any data sent by the host application as a part of the operator operations. This also prevents replay attack as each MAC calculation uses an incrementing counter.
ID_SM_ENC_OUT_OP	Triple-DES 3 SM Encryption key used to encrypt data sent by the Module as a part of the operator operations.
ID_SM_MAC_OUT_OP	Triple-DES 3 SM MAC key used to guarantee integrity on any data sent by the Module as a part of the operator operations. This also prevents replay attack as each MAC calculation uses the host challenge.
ASK_MAC / ASK_ENC	The Applet Start key (ASK) comprises two 3-key Triple-DES keys – ASK_MAC key and ASK_ENC encryption Key. The ASK_MAC Key is used to protect the integrity and authenticate the File System Re-initialization and Change Applet Key services. The ASK_ENC key encrypts the new Key Applet Start Key Set during the Change Applet Key service. It is possible to perform File System Re-Initialization using the Applet Start Key Set.
SEC_AUTH	This CSP is a 3-key Triple-DES MAC key for use in a challenge response protocol. The Secondary Authentication Secret is used as the second level of authentication for cryptographic operations with AES and Triple-DES keys and RSA key pairs.
CH_RSA_KEY_PRIVATE / CH_RSA_KEY_PUBLIC	The eToken Applet Suite implements 0 to n (limited only by available memory) RSA-2048 key pairs used by the CH role in the *Perform Security Operation* service. RSA keys may be used for digital signatures as well as for key decapsulation. However, the key decryption mechanism is not used to establish keys into the module, only to provide key decryption as a service to the caller.
CH_ECDSA_KEY_PRIVATE / CH_ECDSA_KEY_PUBLIC	The eToken Applet Suite implements 0 to n (limited only by available memory) ECDSA key pairs (P-256, P-384 curves) used by the CH role in the *Perform Security Operation* service. ECDSA is used for signature generation.
CH_ECDH_KEY_PRIVATE / CH_ECDH_KEY_PUBLIC	The eToken Applet Suite implements 0 to n (limited only by available memory) ECDH key pairs (P-256, P-384 curves) used by the CH role in the *Perform Security Operation* service. ECDH is used for shared key mechanism
CH_SYMMETRIC	The eToken Applet Suite implements 0 to n (limited only by available memory) AES-128, AES-192, AES-256 or 3-key Triple-DES keys for use by CH role in the *Perform Security Operation* service.

Table 17 – eToken Applet Critical Security Parameters

4.2 USB MCU FW Critical Security Parameters

Key	Description / Usage
ID_FW_DOWNLOAD_RSA_KEY_PUBLIC	2048 bit RSA Public key embedded in the USB MCU FW – used by the FW to validate the new FW signature during FW Download.

Table 18 – USB MCU FW Critical Security Parameters

5 Roles, Authentication and Services

Table 15 lists all operator roles supported by the Module. This Module does not support a maintenance role. The Module clears previous authentications on power cycle. The Module supports GP logical channels, allowing multiple concurrent operators. Authentication of each operator and their access to roles and services is as described in this section, independent of logical channel usage. 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. Authentication data is encrypted during entry (by SD-SDEK), is stored encrypted (by OS-MKDK) and is only accessible by authenticated services.

Role ID	Role Description
CO	(Cryptographic Officer) This role is responsible for card issuance and management of card data via the Card Manager applet. Authenticated using the SCP authentication method with SD-SENC.
CH	Card Holder (User role for FIPS 140-2 validation purposes). The Card Holder role is defined in the context of the eToken Applet Suite and is used to protect keys and data owned by the Card Holder.
ESO	eToken Security Officer This role is responsible for managing the life cycle of the Card Holder (CH).
FSI	File System Initializer This role is responsible for the eToken Applet Suite File System Re-initialization. In additional to the File System initialization, this role is capable of changing the Applet Start Key Set values.
FWU	USB MCU FW Updater; This role is responsible to sign the New FW package with a dedicated RSA 2048 private key to allow updating the 5110 USB MCU FW.
UA	Unauthenticated role

Table 19 - Role Description

5.1 Secure Channel Protocol (SCP) Authentication

The Open Platform Secure Channel Protocol authentication method is performed when the EXTERNAL AUTHENTICATE service is invoked after successful execution of the INITIALIZE UPDATE command. These two commands operate as described next.

The SD-KENC and SD-KMAC keys are used along with other information to derive the SD-SENC and SD-SMAC keys, respectively. The SD-SENC key is used to create a cryptogram; the external entity participating in the mutual authentication also creates this cryptogram. Each participant compares the received cryptogram to the calculated cryptogram and if this succeeds, the two participants are mutually authenticated (the external entity is authenticated to the Module in the CO role).

For SCP03, AES-128, AES-192 or AES-256 keys are used for Global Platform secure channel operations, in which the Module derives session keys from the master keys and a handshake process, performs mutual authentication, and decrypts data for internal use only. The Module encrypts a total of one block (the mutual authentication cryptogram) over the life of the session encryption key; no decrypted data is output by the Module. AES key establishment provides a minimum of 128 bits of security strength. The Module uses the SD-KDEK key to decrypt critical security parameters, and does not perform encryption with this key or output data decrypted with this key.

The strength of GP mutual authentication relies on AES key length, and the probability that a random attempt at authentication will succeed is:

- $\left(\frac{1}{2^{128}}\right)$ for AES 16-byte-long keys;
- $\left(\frac{1}{2^{192}}\right)$ for AES 24-byte-long keys;
- $\left(\frac{1}{2^{256}}\right)$ for AES 32-byte-long keys;

Based on the maximum count value of the failed authentication blocking mechanism, the minimum probability that a random attempt will succeed over a one minute period is 255/2^128.

5.2 eToken Applet Authentication

The ESO or CH authenticates by opening a SM session with the eToken Applet using a challenge response mechanism with the ID_AUTH_OP key, which has an associated error counter in the range one to 15 with default value 15.

The FSI role is authenticated using the ASK_MAC key and utilizing a challenge response mechanism. In all cases, the minimum challenge size is a single 64-bit block, therefore the probability of false authentication is 1/2^64 or approximately 5.4E-20.

For ESO or CH authentication, the error counter limits the probability of false authentication in a one minute period to 15/2^64 or approximately 8.1E-19.

For FSI authentication, the serial communications rate limits the maximum rate of authentication attempts in a one minute period to 1000 attempts, therefore the probability of false authentication is 1000/2^64 or approximately 5.4E-17.

5.3 FW Updater Authentication

The FW updater utilizes RSA 2048-bit signature verification to authenticate new signed firmware to be loaded. According to NIST SP 800-57 and NIST SP 800-131A Rev 1, the RSA 2048-bit signature verification method provides an estimated security strength of 112-bits.

5.4 Platform Services

All services implemented by the Module are listed in the tables below. Each service description also describes all usage of CSPs by the service.

Service	Description
Card Reset (Self-test)	Power cycle the Module by removing and reinserting it into the contact reader slot, or by reader assertion of the RST signal. The *Card Reset* service will invoke the power on self-tests described in Section §10-Self-test. Moreover, on any card reset, the Module overwrites with zeros the RAM copy of, OS-RNG-STATE, SD-SENC, SD-SMAC and SD-SDEK. The Module can also write the values of all CSPs stored in EEPROM as a consequence of restoring values in the event of card tearing or a similar event. During the self-tests, the module generates the RAM copy of OS-RNG-STATE and updates the EEPROM copy of OS-RNG-STATE.
EXTERNAL AUTHENTICATE	Authenticates the operator and establishes a secure channel. Must be preceded by a successful INITIALIZE UPDATE. Uses SD-SENC and SD-SMAC.
INITIALIZE UPDATE	Initializes the Secure Channel; to be followed by EXTERNAL AUTHENTICATE. Uses the SD-KENC, SD-KMAC and SD-KDEK master keys to generate the SD-SENC, SD-SMAC and SD-SDEK session keys, respectively.
GET DATA	Retrieve a single data object. Optionally uses SD-SENC, SD-SMAC (SCP).
MANAGE CHANNEL	Open and close supplementary logical channels. Optionally uses SD-SENC, SD-SMAC (SCP).
SELECT	Select an applet. Does not use CSPs.

Table 20 - Unauthenticated Services and CSP Usage

Service	Description	CO
DELETE	Delete an applet from EEPROM. This service is provided for the situation where an applet exists on the card, and does not impact platform CSPs. Optionally uses SD-SENC, SD-SMAC (SCP).	X
GET STATUS	Retrieve information about the card. Does not use CSPs. Optionally uses SD-SENC, SD-SMAC (SCP).	X
INSTALL	Perform Card Content management. Optionally uses SD-SENC, SD-SMAC (SCP). Optionally, the Module uses the DAP-SYM key to verify the package signature.	X
LOAD	Load a load file (e.g., an applet). Optionally uses SD-SENC, SD-SMAC (SCP).	X
PUT DATA	Transfer data to an application during command processing. Optionally uses SD-SENC, SD-SMAC (SCP).	X
PUT KEY	Load Card Manager keys The Module uses the SD-KDEK key to decrypt the keys to be loaded. Optionally uses SD-SENC, SD-SMAC (SCP).	X
SET STATUS	Modify the card or applet life cycle status. Optionally uses SD-SENC, SD-SMAC (SCP).	X
STORE DATA	Transfer data to an application or the security domain (ISD) processing the command. Optionally, updates OS-GLOBALPIN. Optionally uses SD-SENC, SD-SMAC (SCP).	X
GET MEMORY SPACE	Monitor the memory space available on the card. Optionally uses SD-SENC, SD-SMAC (SCP).	X
SET ATR	Change the card ATR. Optionally uses SD-SENC, SD-SMAC (SCP).	X

Table 21 – Authenticated Card Manager Services and CSP Usage

All of the above commands use the SD-SENC and SD-SMAC keys for secure channel communications, and SD-SMAC for firmware load integrity.

The card life cycle state determines which modes are available for the secure channel. In the SECURED card life cycle state, all command data must be secured by at least a MAC. As specified in the GP specification, there exist earlier states (before card issuance) in which a MAC might not be necessary to send Issuer Security Domain commands. Note that the LOAD service enforces MAC usage.

5.5 eToken Applet Services

All services implemented by the eToken Applet 1.8 applet are listed in the table below.

All services that are related to Key import/gen or Key Crypto Operations are Secure Messaging Protected.

Service	Description	ESO	CH	FSI	UA
Operator Logon	Authenticate the CH or ESO role. Uses ID_AUTH_OP key from Secure Messaging Key Set to authenticate operator.	X	X		
Credential Change	Change the current Secure Messaging Key Set. Uses ID_SM_ENC_IN_OP and ID_SM_MAC_IN_OP from Secure Messaging Key Set to decrypt and verify MAC value on new credentials.	X	X		
CH Unlocking	Unlock the CH Key Set. The ESO provides the new CH key set. All CH data and other keys remain untouched. Uses ID_SM_ENC_IN_OP and ID_SM_MAC_IN_OP from Secure Messaging Key Set to decrypt and verify MAC value on new CH key set.	X			
File System Re-initialization	The data in the SafeNet eToken Applet are cleared and the eToken file system is re-initialized. Deleting all keys stored in the eToken file system. This service executed from the FSI role. ASK_ENC key from Applet Start Key set is used to authenticate FSI role.			X	
Generate CH RSA Key Pair	Generate a CH RSA Key Pair. Generates RSA Key Pair, including CH_RSA_KEY_PRIVATE and CH_RSA_KEY_PUBLIC keys.		X		
Generate CH ECDSA Key Pair	Generate a CH ECDSA Key Pair, including CH_ECDSA_KEY_PRIVATE and CH_ECDSA_KEY_PUBLIC keys.		X		
Generate CH ECDH Key Pair	Generate a CH ECDH Key Pair, including CH_ECDSA_KEY_PRIVATE and CH_ECDSA_KEY_PUBLIC keys.		X		

Create Secondary Authentication Secret	Import Secondary Authentication Secret SEC_AUTH. Updates SEC_AUTH. Uses ID_SM_ENC_IN_OP and ID_SM_MAC_IN_OP from Secure Messaging Key Set to decrypt and authenticate the new value.		X		
Perform Security Operation	Use a CH RSA Key Pair to generate/verify signatures or decryption of symmetric keys. Use a CH ECDSA Key Pair to generate, compute/verify signatures or compute a shared secret. Use a CH ECDH Key pair for shared key mechanism. Use AES and Triple-DES keys for encryption/decryption. Key decryption does not establish a CSP into the Module. Uses CH_RSA_KEY_PRIVATE, CH_RSA_KEY_PUBLIC, CH_ECDSA_KEY_PRIVATE, CH_ECDSH_KEY_PRIVATE, CH_ECDSA_KEY_PUBLIC, CH_ECDH_KEY_PUBLIC or CH_SYMMETRIC keys, depending on operation type.		X		
Store and read data	Store and read data objects. Uses the Secure Messaging key set to encrypt/decrypt and authenticate data on input/output.		X		
Import RSA\ECC key pair and import TDEA \AES symmetric keys	Import RSA\ECC key pair. Writes CH_RSA_PRIVATE_KEY, CH_RSA_PUBLIC_KEY or CH_ECDSA_PRIVATE_KEY, CH_ECDSA_PUBLIC_KEY or CH_ECDH_PRIVATE_KEY, CH_ECDH_PUBLIC_KEY or CH_SYMMETRIC Uses the Secure Messaging key set to encrypt/decrypt and authenticate keys on input/output.		X		
Manage file system	Create files, delete files, admin files, list directories, resize filesystem, wipe filesystem. Uses the Secure Messaging key set to encrypt/decrypt and authenticate commands on input/output.	X	X		
Manage objects	Admin object, get object info, list objects. Uses the Secure Messaging key set to encrypt/decrypt and authenticate commands on input/output.	X	X		

Export public key	Export public key. The service does not utilize any CSPs.				X
Set/read volatile data	Set or read application-specific volatile data for the applet.				X

Table 22 – eToken Applet Services and CSP Usage

5.6 USB MCU FW Services

All services implemented by the USB MCU FW are listed in the table below.

Service	Description	UA	FWU
USB	This module provides framework for the USB Standard and Class requests, such as VSR, CCID protocols, to allow either ISO7816 communication with the SC or Commands which are directed to the FW such as FW Update, FW get Info, etc.	X	
FW Update	This module defines an interface for firmware update process; FW is protected by an RSA 2048 signature. The signature verification is for the purposes of authenticating the USB FW download.		X
Crypto Module	This module includes the API dedicated to using SHA-256 and RSA-2048 verify (RSA PKCS#1-v1_5 format).	X	

Table 23 – USB MCU FW applet Services

5.7 eTPnP Applet Services

In addition to the authenticated services, the module provides the Minidriver Information Service when the Minidriver eTPnP Applet is selected. This service is available without authentication. It provides non-security relevant information used by the host operating system to recognize the module.

The Minidriver Information Service does not utilize any CSPs.

Service	Description	UA
GET DATA	Retrieves the following information: • GUID	X

Table 24 – eTPnP applet Services

6 Finite State Model

The CM is designed using a finite state machine model that explicitly specifies every operational and error state.

The CM includes Power on/off states, Cryptographic Officer states, User services states, applet loading states, Key/PIN loading states, Self-test states, Error states, and the GP life cycle states.

An additional document (Finite State Machine document) identifies and describes all the states of the module including all corresponding state transitions.

7 Physical Security Policy

eToken 5110 is a multiple-chip standalone cryptographic module. Two (2) ICs are mounted on a PCB assembly with a connector and passive components, covered by epoxy on both sides, exposing only the LED and USB connector. The Module is intended to be covered within a plastic enclosure. Physical inspection inside the Module boundary is not practical, as the epoxy layer is opaque.

8 Operational Environment

This section does not apply to CM. No code modifying the behavior of the CM operating system can be added after its manufacturing process.

Only authorized applets can be loaded at post-issuance under control of the Cryptographic Officer. Their execution is controlled by the CM operating system following its security policy rules.

9 Electromagnetic Interference and Compatibility (EMI/EMC)

The Module conforms to the EMI/EMC requirements specified by part 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B.

10 Self-test

10.1 Power-on Self-test

Each time the CM is powered up it tests that the cryptographic algorithms still operate correctly and that sensitive data have not been damaged. Power-on self–tests are available on demand by power cycling the CM.

On power-on or reset, the CM performs the self-tests described in table below. All KATs must be completed successfully prior to any other use of cryptography by the CM. If one of the KATs fails, the CM enters the Card Is Mute error state.

Test Target	Description
Firmware Integrity	16 bit CRC performed over all code located in Flash memory (for OS and Applets).
DRBG	Performs DRBG SP 800-90 Section 11.3 instantiate and generate health test KAT with fixed inputs (no derivation function and no reseeding supported)
Triple-DES	Performs separate encrypt and decrypt KATs using 3-Key Triple-DES in ECB mode.
AES	Performs decrypt KAT using an AES 128 key in ECB mode. AES encrypt is self-tested as an embedded algorithm of AES-CMAC.
KBKDF AES-CMAC	Performs a KDF AES-CMAC KAT using an AES 128 key and 32-byte derivation data. The KAT computes session keys and verifies the result. Note that KDF KAT is identical to an AES-CMAC KAT; the only difference is the size of input data.
RSA	Performs separate RSA PKCS#1 signature and verification KATs using an RSA 2048 bit key, and a RSA PKCS#1 signature KAT using the RSA CRT implementation with a 2048 bit key.
ECC CDH	Performs an ECC CDH KAT using an ECC P-224 key.(same crypto engine than for ECDSA KAT)
SHA-1	Performs a SHA-1 KAT.
SHA-256	Performs a SHA-256 KAT.
SHA-512	Performs a SHA-512 KAT.
USB MCU FW Integrity	32 bit CRC performed over all FW executable code, located in MCU Flash memory. 32 bit CRC performed over FW configuration block, located in MCU Flash memory.
USB MCU FW RSA Signature Verification	Perform RSA 2048 PKCS#1 v1.5 Sig Verification KAT.
USB MCU FW SHA-256	Performs a SHA-256 KAT.

Table 25 – Power-On Self-Test

10.2 Conditional Self-tests

On every call to the [SP 800-90] DRBG, the CM performs the FIPS 140-2 Continuous RNG test to assure that the output is different than the previous value.

When any asymmetric key pair is generated (for RSA or ECC keys) the CM performs a pair-wise consistency test.

When new firmware is loaded into the CM using the LOAD command, the CM verifies the integrity and authenticity of the new firmware (applet) using the SD-SMAC key for MAC process.

USB MCU FW design includes a firmware load service to support necessary updates. Updatable MCU FW code is signed by RSA-2048 SHA-256 private key to avoid non-authorized FW update.

When new MCU FW is loaded into the USB MCU, the MCU FW verifies the integrity and authenticity of the new firmware using the RSA Signature verify method.

The RSA-2048 Modulus and Public Exponent data are hardcoded in firmware code. Firmware does not provide interface to change or read this key.

11 Design Assurance

The CM meets the Level 3 Design Assurance section requirements.

11.1 Configuration Management

An additional document (Configuration Management Plan document) defines the methods, mechanisms and tools that allow to identify and place under control all the data and information concerning the specification, design, implementation, generation, test and validation of the card software throughout the development and validation cycle.

11.2 Delivery and Operation

Some additional documents ('Delivery and Operation', 'Reference Manual', 'Card Initialization Specification' documents) define and describe the steps necessary to deliver and operate the CM securely.

11.3 Guidance Documents

The Guidance document provided with CM is intended to be the 'Reference Manual'. This document includes guidance for secure operation of the CM by its users as defined in the section: Roles, Authentication and Services.

11.4 Language Level

The CM operational environment is implemented using a high level language. A limited number of software modules have been written in assembler to optimize speed or size.

The eToken Applet is a Java applet designed for the Java Card environment.

12 Mitigation of Other Attacks Policy

The Module implements defenses against:

- Fault attacks
- Side channel analysis (Timing Analysis, SPA/DPA, Simple/Differential Electromagnetic Analysis)
- Probing attacks
- Card tearing

13 Security Rules and Guidance

The Module implementation also enforces the following security rules:

- No additional interface or service is implemented by the Module which would provide access to CSPs.
- Data output is inhibited during key generation, self-tests, zeroization, and error states.
- There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
- The Module does not support manual key entry, output plaintext CSPs or output intermediate key values.
- Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.
- In accordance to NIST guidance, operators are responsible for ensuring that a single Triple-DES key shall not be used to encrypt more than 2^{16} 64-bit data blocks.

END OF DOCUMENT