Giesecke & Devrient

# StarSign Crypto-USB Token S powered by Sm@rtCafé Expert 7.0 Secure Element

# FIPS 140-2 Cryptographic Module Non-Proprietary Security Policy

| | |
|---|---|
| Author | Giesecke & Devrient |
| Status | Final |
| Edition | 12 January 2015, 1.0 |

Giesecke & Devrient GmbH
Prinzregentenstraße 159
P.O. Box 80 07 29
D-81607 Munich

# Table of Contents

## List of Tables

## List of Figures

# References

| Acronym | Full Specification Name |
|---|---|
| [FIPS140-2] | NIST, *Security Requirements for Cryptographic Modules*, May 25, 2001 |
| [GlobalPlatform] | *GlobalPlatform Consortium: GlobalPlatform Card Specification 2.2.1*, January 2011, http://www.globalplatform.org<br>*GlobalPlatform Consortium: GlobalPlatform Card Specification 2.2* Amendment A, Confidential Card Content Management, Version 1.0, October 2007 |
| [ISO 7816] | ISO/IEC 7816-1:2011 *Identification cards -- Integrated circuit(s) cards with contacts -- Part 1: Physical characteristics*<br>ISO/IEC 7816-2:2007 *Identification cards - Integrated circuit cards - Part 2: Dimensions and location of the contacts. ISO/IEC 7816-2:2007.*<br>ISO/IEC 7816-3:2006 *Identification cards - Integrated circuit cards - Part 3: Electrical interface and transmission protocols. ISO/IEC 7816-3:2006.*<br>ISO/IEC 7816-4:2013 *Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange.* |
| [JavaCard] | *Java Card 3 Platform Runtime Environment (JCRE) Specification, Classic Edition. Version 3.0.4*<br>*Java Card 3 Platform Virtual Machine (JCVM) Specification, Classic Edition. Version 3.0.4*<br>*Java Card 3 Platform Application Programming Interface, Classic Edition. Version 3.0.4*<br>Published by Oracle, September 2011 |
| [SP800-131A] | *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011 |
| [ANS X9.31] | American Bankers Association, *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*, ANSI X9.31-1998 - Appendix A.2.4. |
| [SP 800-67] | NIST Special Publication 800-67, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, version 1.2, July 2011 |
| [FIPS113] | NIST, *Computer Data Authentication*, FIPS Publication 113, 30 May 1985. |
| [FIPS197] | NIST, *Advanced Encryption Standard (AES)*, FIPS Publication 197, November 26, 2001. |
| [PKCS#1] | *PKCS #1 v2.1: RSA Cryptography Standard*, RSA Laboratories, June 14, 2002 |
| [FIPS 186-4] | NIST, *Digital Signature Standard (DSS)*, FIPS Publication 186-4, July, 2013 |
| [SP 800-56A] | NIST Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, March 2007 |
| [FIPS 180-4] | NIST, *Secure Hash Standard*, FIPS Publication 180-4, March 2012 |
| [SP800-108] | NIST, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, October 2009 |
| [SP800-38F] | NIST, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, December 2012 |
| [IG] | NIST, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*, last updated 2 March 2015. |
| [RS] | Irving S. Reed, Gustave Solomon: *Polynomial codes over certain finite fields*. In: Journal of the Society for Industrial and Applied Mathematics, SIAM J. 8, 1960, ISSN 0036-1399, p. 300–304. |

**Table 1 – References**

## Acronyms and definitions

| Acronym | Definition |
|---|---|
| APDU | Application Protocol Data Unit, see [ISO 7816]. A logical data packet. |
| API | Application Programming Interface |
| ATR | Answer To Reset |
| CSP | Critical Security Parameter, see [FIPS 140-2] |
| DAP | Data Authentication Pattern, see [GlobalPlatform] |
| DPA | Differential Power Analysis |
| GP | GlobalPlatform |
| IC | Integrated Circuit |
| ISD | Issuer Security Domain, see [GlobalPlatform] |
| KAT | Known Answer Test |
| NVM | Non-volatile memory |
| PCT | Pairwise Consistency Test |
| QFN | Quad Flat Non-Leaded Package |
| SCP | Secure Channel Protocol, see [GlobalPlatform] |
| SPA | Simple Power Analysis |

**Table 2 – Acronyms and Definitions**

# 1 Introduction

This document defines the Security Policy for the Giesecke & Devrient StarSign Crypto-USB Token S powered by Sm@rtCafé Expert 7.0 Secure Element cryptographic module, hereafter denoted *the module*. The module, validated to FIPS 140-2 overall Level 3, is a single chip module implementing the GlobalPlatform operational environment, with Card Manager and a Demonstration Applet.

The Demonstration Applet is available only to demonstrate the complete cryptographic capabilities of the module for FIPS 140-2 validation, and is not intended for general use. The term *platform* herein is used to describe the chip and operational environment, not inclusive of the Demonstration Applet.

The module is a limited operational environment under the FIPS 140-2 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 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 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**

## 1.1 Versions, Configurations and Modes of operation

**Hardware:** SLE78CUFX5000PH (M7893 B11)
**Firmware:** Sm@rtCafé Expert 7.0, Demonstration Applet V1.0
**Packaging:** P/PG-VQFN-32-13

The chip and firmware are identical in all configurations. The chip design is a superset of all possible interface options; unused options are disabled during production.

The module is always issued in Approved mode; the explicit indicator of Approved mode is given in the ATR: the value 0x46 (‘F’) in Historical Byte 9 indicates the Approved mode.

```
interface bytes                    historical bytes
3B F9 96 00 00 81 31 FE 45         53 43 45 37 20 0E 00 20 [46]
                                   S  C  E  7              F
```

## 2 Hardware and Physical Cryptographic Boundary

The module hardware and physical cryptographic boundary of the module is depicted in Figure 1. The cryptographic boundary is the surface and edges of the package as shown in the figure below (depicted by the red line). The module may be used in two configurations:

- ISO 7816 embedded SMD, denoted “SMD” in this document. In this configuration, intended for use in mobile devices, or similar end use devices, the module provides a fully compliant ISO/IEC 7816 contact interface.
- A USB Token configuration. In this configuration, intended for use in USB tokens, the module provides a fully compliant USB 2.0 interface.

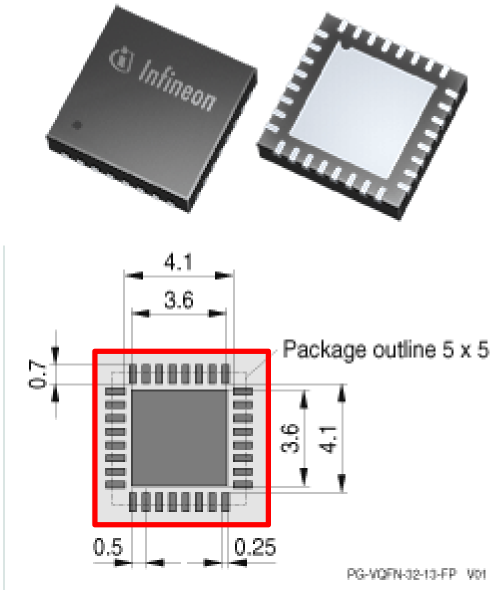

**Figure 1 – Hardware and physical cryptographic boundary of the module**

Figure 2 shows how the module is used in a USB Token configuration. In this case the module is soldered on a PCB and is connected to the USB connector via USB interface. The PCB is enclosed by a token housing. The module may also be connected via an LED of the token via a general purpose output line.

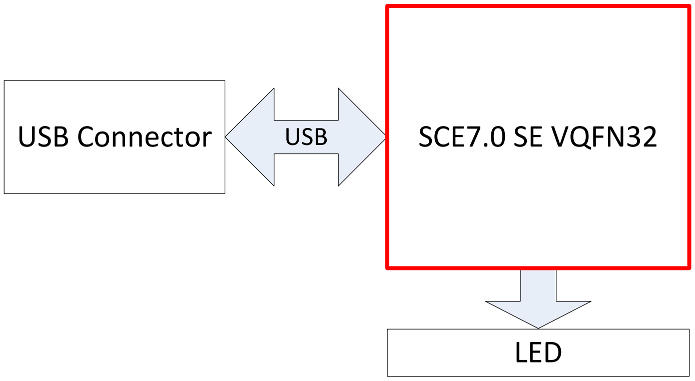

**Figure 2 – Module in a USB Token Configuration**

Figure 3 shows how the module in a USB Token configuration could be mounted in a USB housing.

**Figure 3 – Example of the USB form factor**

Figure 4 shows the layout of the module with the different ports and interfaces.

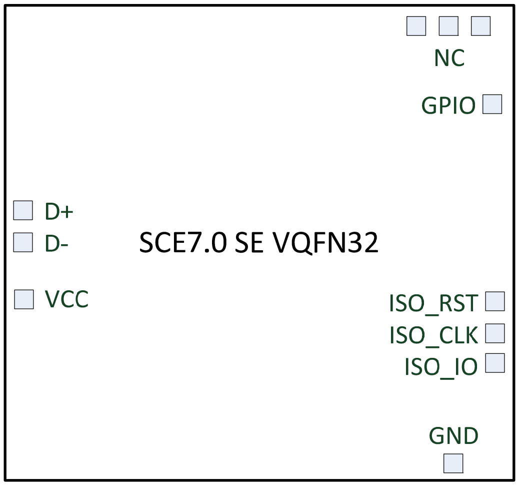

**Figure 4 – Layout of Module**

Table 4 describes the different ports of the module.

| Port | Description | Logical Interface Type |
|---|---|---|
| GND | Ground | Power |
| VCC | Supply voltage | Power |
| D+, D- | USB Input/output | Control in, Data in, Data out, Status out (USB Token configuration only) |
| ISO_IO | ISO 7816: Input/output | Control in, Data in, Data out, Status out (SMD configuration only) |
| ISO_RST | ISO 7816: Reset | Control-in (SMD configuration only) |
| ISO_CLK | ISO 7816: Clock | Control-in (SMD configuration only) |
| USBCAP | Connected to GND for USB operation | External buffer capacitor (USB configuration only) |
| GPIO | Signals the token LED. (output only) | General purpose output (USB configuration only) |
| NC | Not connected | |

**Table 4 – Ports and Interfaces**

In the *USB Token* configuration the module uses the USB interface for control/data input and status/data output. In the *SMD* configuration the module uses the ISO 7816 interface for control/data input and status/data output. ISO/IEC 7816-4 APDUs (logical data packets) are used for communication in both module configurations, SMD and USB Token.

## 2.1 Firmware and Logical Cryptographic Boundary

Figure 5 depicts the module operational environment.

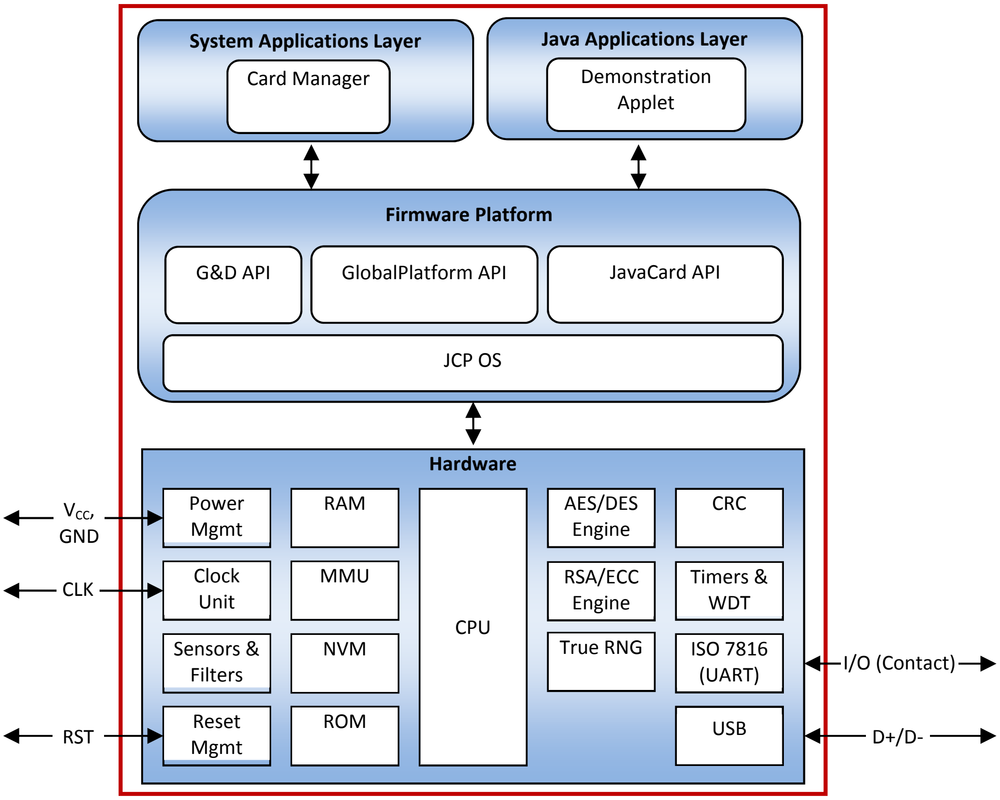

**Figure 5 – Module Block Diagram**

The JavaCard, GlobalPlatform and G&D APIs are internal interfaces available only to applets and security domains (i.e., Card Manager). Only applet services are available at the card edge (the interfaces that cross the cryptographic boundary). Section 3 describes applet functionality in greater detail.

The NVM is separated into segments with different access rules, enforced by the hardware MMU. The MMU is initialized with the correct settings by startup code, and verified by the operating system each time the system starts. The MMU settings cannot be changed at run time. All code is executed from ROM and NVM.

# 3 Cryptographic Functionality

The module implements the Approved and Non-Approved but Allowed cryptographic functions listed in Tables 5 and 6 below.

| Algorithm | Description | Cert # |
|---|---|---|
| DRBG | [SP 800-90A] AES-256 CTR_DRBG; without prediction resistance. | 455 |
| Triple-DES | [SP 800-67] Triple Data Encryption Algorithm. The module supports 3-Key keys only, and CBC and ECB modes. | 1637 |
| Triple-DES MAC | [FIPS113] Triple-DES MAC, vendor affirmed based on Cert. 1637. | 1637 |
| AES | [FIPS 197] Advanced Encryption Standard algorithm. The module supports AES-128, AES-192- and AES-256 keys, and ECB and CBC modes. | 2721 |
| AES CMAC | [SP800-38B] AES-256 CMAC. The module supports AES-128, AES-192 and AES-256 keys. | 2720 |
| SHA-1 | SHA-1 | 2290 |
| SHA-2 | [FIPS 180-4] Secure Hash Standard compliant one-way (hash) algorithms; SHA-224, SHA-256, SHA-384, SHA-512 | 2289 |
| SHA-2 | SHA-256 | 2288 |
| RSA | [FIPS 186-4] RSA key generation, signature generation and verification. The module supports 2048-bit RSA keys. | 1506 |
| RSA CRT | [FIPS 186-4] RSA key generation and signature generation. The module supports 2048-bit RSA keys. | 1507 |
| DSA | [FIPS 186-4] DSA key generation, signature generation and verification. The module supports 2048 bit keys. | 837 |
| ECDSA | [FIPS 186-4] Elliptic Curve Digital Signature Algorithm. The module supports the NIST defined P-224, P-256, P-384, P-521 curves for key pair generation, signature and signature verification. | 476 |
| KDF | [SP 800-108] CMAC-based KDF with AES-128, AES-192, AES-256. | 18 |
| CVL | [SP 800-56A] The Section 5.7.1.2 ECC CDH Primitive only (as used by the PIV specification). The module supports the NIST defined P-224 P-256, P-384 and P-521 curves. | 177 |

**Table 5 – Approved Cryptographic Functions**

| Algorithm | Description |
|---|---|
| TRNG | Hardware TRNG used to seed the FIPS approved DRBG. |
| Key Wrap | AES (Cert.#2721, key wrapping; key establishment methodology provides 128 to 256 bits of encryption strength) |

**Table 6 – Non-Approved but Allowed Cryptographic Functions**

## 3.1 Critical Security Parameters and Public Keys

All CSPs and public keys used by the module are described in this section. In the tables below, the OS prefix denotes operating system, the SD prefix denotes the GlobalPlatform Security Domain, the DAP prefix denotes the GlobalPlatform Data Authentication Protocol, and the DEM prefix denotes a Demonstration Applet CSP.

| CSP | Description / Usage |
|---|---|
| OS-RNG-STATE | 384 bit value; the current RNG state. |
| SD-KENC | AES-128, AES-192, AES-256 Master key used to generate SD-SENC. |
| SD-KMAC | AES-128, AES-192, AES-256 Master key used to generate SD-SMAC. |
| SD-KDEK | AES-128, AES-192, AES-256 Sensitive data decryption key used to decrypt CSPs. |
| SD-SENC | AES-128, AES-192, AES-256 Session encryption key used to encrypt / decrypt secure channel data. |
| SD-SMAC | AES-128, AES-192, AES-256 Session MAC key used to verify inbound secure channel data integrity. |
| SD-SRMAC | AES-128, AES-192, AES-256 Session MAC key used to verify response secure channel data integrity. |
| DAP-SYM | AES-128, AES-192, AES-256 authentication key used by the *Manage Content* service. |
| DEM-AUTH | An 8 byte PIN value allowing all 256 values for each byte, used by the *PIN Authentication* service. The module always checks all 8 bytes of the PIN. |
| DEM-KAP-PRI | EC P-256 private key used to demonstrate the ECC CDH shared secret generation. The *Key Agreement Primitive* service allows any of the valid EC curves to be used. |
| DEM-MAC | 3-Key Triple-DES ENC or MAC key used by the *Message Authentication* service. |
| DEM-SGV-PRIV | DSA 2048 bit, ECDSA P-256 or RSA 2048 bit private key used by the *Digital Signature* service. |

**Table 7 – Critical Security Parameters**

| Key | Description / Usage |
|---|---|
| DAP-PUB | RSA 2048 new firmware signature verification key. |
| DEM-KAP-PUB | EC P-256 ECDSA public key used by the *Key Agreement Primitive* service. |
| DEM-SGV-PUB | DSA 2048 bit, EC P-256 ECDSA or RSA 2048 bit public key used by the *Digital Signature* service. |

**Table 8 – Public Keys**

# 4 Roles, Authentication and Services

The module:

- Does not support a maintenance role.
- Clears previous authentications on power cycle.
- Supports GlobalPlatform SCP logical channels, allowing concurrent operators in a limited fashion.

Authentication of each operator and their access to roles and services is as described below, 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-KDEK), and is only accessible by authenticated services.

Table 9 lists all operator roles supported by the module.

| Role ID | Role Description |
|---|---|
| CO | Cryptographic Officer – role that manages module content and configuration, including issuance and management of module data via the ISD. Authenticated as described in *Secure Channel Protocol Authentication* in Section 4.1 below. |
| User | User – role for use in Demonstration applet. Authenticated as described in *Demonstration Applet Authentication* in Section 4.2 below. |

**Table 9 – Roles Supported by the Module**

## 4.1 Secure Channel Protocol 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 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).

The probability that a random attempt will succeed using this authentication method is:

- 1/2^128 = 2.9E-39 (for any of AES-128/192/256 SD-KENC/SD-SENC, assuming a 128-bit block)

The module enforces a maximum of fifteen (15) consecutive failed SCP authentication attempts. The probability that a random attempt will succeed over a one minute interval is:

- 15/2^128 = 4.4E-38 (for any of AES-128/192/256 SD-KENC/SD-SENC, assuming a 128-bit block)

## 4.2 Demonstration Applet Authentication Method

The Demonstration Applet Authentication method is provided by the *Secure Channel* service combined with the *Authenticate* service. The module accepts an 8 byte PIN value and compares all 8 bytes to a stored reference, with no restriction on character space (each character can be any value from 0-255). The probability that a random attempt will succeed using this authentication method is:

- 1/256^8 = 5.4E-20

The module enforces a maximum of three (3) consecutive failed authentication attempts. The probability that a random attempt will succeed over a one minute interval is:

- 3/256^8 = 1.6E-19.

## 4.3 Services

All services implemented by the module are listed in the tables below.

| Service | Description |
|---|---|
| Context | Selects an applet or manage logical channels. |
| Module Info (Unauthenticated) | Reads unprivileged data objects, e.g., module configuration or status information. |
| Module Reset | Power cycles or resets the module. Includes Power-On Self-Test. |

**Table 10 – Unauthenticated Services**

| Service | Description | CO | User |
|---|---|---|---|
| Lifecycle | Modifies the card or applet life cycle status. | X | |
| Manage Content | Loads and installs application packages and associated keys and data. | X | |
| Module Info (Authenticated) | Reads module configuration or status information (privileged data objects). | X | |
| Secure Channel | Establishes and uses a secure communications channel. | X | X |
| PIN Authentication | Demonstrates PIN authentication with OwnerPIN. | | X |
| Manage Applet Content | Creates uninitialized key objects for use by the demo applet's cryptographic services. Deletes on-card key objects, arrays, signature objects. | | X |
| Keys | Generates keys and initializes symmetric and asymmetric key objects for the cryptographic services. | | X |
| Digital Signature | Demonstrates DSA, RSA, and ECDSA digital signature generation and verification. | | X |
| Key Agreement Primitive | Demonstrates Approved ECC CDH primitive (SP 800-56A Section 5.7.1.2). | | X |
| Message Authentication | Demonstrates Triple-DES encryption, decryption and MAC. | | X |
| Message Digest | Demonstrates secure message digest (hash) generation (SHA-224, SHA-256, SHA-384, and SHA-512). | | X |

**Table 11 – Authenticated Services**

| | CSPs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Service** | **OS-RNG-STATE** | **SD-KENC** | **SD-KMAC** | **SD-KDEK** | **SD-SENC** | **SD-SMAC** | **SD-SRMAC** | **DAP-SYM** | **DEM-AUTH** | **DEM-KAP-PRI** | **DEM-MAC** | **DEM-SGV-PRIV** |
| Context | -- | -- | -- | -- | Z | Z | Z | -- | -- | -- | -- | -- |
| Module Info (Unauthenticated) | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| Module Reset | GEW | -- | -- | -- | Z | Z | Z | -- | -- | -- | -- | -- |
| Lifecycle[1] | Z | Z | Z | Z | E | E | E | Z | Z | Z | Z | Z |
| Manage Content[2] | -- | W | W | W | E | E | E | EW | Z | Z | Z | Z |
| Module Info (Authenticated) | -- | -- | -- | -- | E | E | E | -- | -- | -- | -- | -- |
| Secure Channel | EW | E | E | -- | GE | GE | GE | -- | -- | -- | -- | -- |
| PIN Authentication | -- | -- | -- | -- | E | E | -- | -- | E | -- | -- | -- |
| Manage Applet Content | -- | -- | -- | -- | E | E | -- | -- | -- | C | C | C |
| Keys | EW | -- | -- | -- | E | E | -- | -- | -- | GZ | GZ | GZ |
| Digital Signature | EW | -- | -- | -- | E | E | -- | -- | -- | -- | -- | GE |
| Key Agreement Primitive | EW | -- | -- | -- | E | E | -- | -- | -- | GE | -- | -- |
| Message Authentication | -- | -- | -- | -- | E | E | -- | -- | -- | -- | E | -- |
| Message Digest | -- | -- | -- | -- | E | E | -- | -- | -- | -- | -- | -- |

**Table 12 – Access to CSPs by Service**

- G = Generate: The module generates the CSP.
- C = Create: The module uninitializes key objects for signature and cipher algorithms.
- R = Read: The module reads the CSP (read access to the CSP by an outside entity).
- E = Execute: The module executes using the CSP.
- W = Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module or when the module overwrites an existing CSP.
- Z = Zeroize: The module zeroizes the CSP. For the Context service, SD session keys are destroyed on applet deselect (channel closure)
- -- = Not accessed by the service.

[1] Zeroize in this row corresponds to card termination.

[2] Zeroize in this row corresponds to the Demonstration Applet deletion.

# 5 Self-test

## 5.1 Power-On Self-tests

On power-on or reset, the module performs self-tests as described in Table 13 below. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fails, the system emits an error code (0x6666) and enters the SELF-TEST ERROR state.

| Test Target | Description |
|---|---|
| Firmware Integrity | 16 bit Reed-Solomon EDC performed over all code in the cryptographic boundary. |
| DRBG | Performs a fixed input KAT. |
| Triple-DES | Performs separate encrypt and decrypt KATs using 3-Key Triple-DES in ECB mode. |
| AES | Performs a decrypt KAT using an AES-128 key in ECB mode. |
| SP 800-108 KDF | Performs a KAT of SP 800-108 KDF. This self-test is inclusive of AES CMAC and AES encrypt function self-test. |
| RSA | Performs separate RSA signature and verify KATs using an RSA 2048-bit key. |
| RSA CRT | Performs RSA CRT signature KATs using an RSA 2048-bit key. |
| ECDSA | Performs pairwise consistency test using the P-521 curve. |
| SHA-1 | Performs a fixed input KAT. |
| SHA-256 | Performs a fixed input KAT. |
| SHA-256 (2) | Performs a fixed input KAT for the 2nd SHA-256 implementation. |
| SHA-512 | Performs a fixed input KAT. |
| DSA | Performs a pairwise consistency test using a DSA 2048-bit key. |
| ECC CDH | Primitive “Z” Computation KAT for [SP 800-56A] Section 5.7.1.2 ECC CDH Primitive using the P-521 curve. |

**Table 13 – Power-On Self-Test**

## 5.2 Conditional Self-tests

On every call to the DRBG, the module performs the AS09.42 continuous RNG test to assure that the output is different than the previous value.  If the continuous RNG test fails, the module enters the SELF-TEST ERROR state. The TRNG hardware includes a continuous comparison test, such that each word formed is compared to the previous value; a duplicate value is discarded, and the TRNG status indicates not ready.

When an RSA, DSA or ECDSA key pair is generated the module performs a pairwise consistency test. If the pairwise consistency test fails, the module enters the SELF-TEST ERROR state.

When new firmware is loaded into the module using the *Manage Content* service, the module verifies the integrity of the new firmware (applet) using MAC verification with the SD-SMAC key. Optionally, the module may also verify a signature of the new firmware (applet) using the DAP-SV-PUB public key or the DAP-SYM key; the signature block in this scenario is generated by an external entity using the private key corresponding to DAP-SV-PUB or the symmetric DAP-SYM. Failure to verify the new firmware results in the BAD APDU error state; the module returns an error specific to the situation (MAC failure or DAP failure).

# 6 Physical Security Policy

The module is a single-chip implementation available in SMD packaging that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. The module was tested at ambient temperature only.

The module is intended to be mounted in additional packaging as described in Section 2. Physical inspection of the die for tamper evidence is typically not practical after packaging.

# 7 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.

# 8 Mitigation of Other Attacks Policy

The module implements defenses against:

- Physical fault induction, such as laser, light, clock glitch or similar attacks
- Side-channel attacks (SPA/DPA and timing analysis)
- Differential fault analysis (DFA)

# 9 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.