NXP Semiconductors
SE050

FIPS 140-2 Cryptographic Module
Non-Proprietary Security Policy

Document Version: 1.3
Date: 2/1/2021

Table of Contents

List of Tables

List of Figures

References

Table 1: References

Acronym	Full Specification Name
References used in Approved Algorithms Table	
[38A]	NIST, Special Publication 800-38A, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, December 2001
[38ACS]	NIST, Special Publication 800-38A Addendum, *Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode*, October 2010
[38B]	NIST, Special Publication 800-38B, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*, May 2005
[38F]	NIST, Special Publication 800-38F, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, December 2012
[56A]	NIST, Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, March 2007
[56Arev3]	NIST, Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, Revision 3, April 2018
[56B]	NIST, Special Publication 800-56B*, Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography*, Revision 1, September 2014
[56Brev2]	NIST, Special Publication 800-56B*, Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography*, Revision 2, March 2019
[56Crev1]	NIST, Special Publication 800-56C*, Recommendation for Key-Derivation Methods in Key-Establishment Schemes*, Revision 1, April 2018
[67]	NIST Special Publication 800-67, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, Revision 2, July, 2017
[90A]	NIST, Special Publication 800-90A, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, Revision 1, June, 2015
[108]	NIST, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, FIPS Publication 108, October, 2009
[133]	NIST Special Publication SP800-133, *Recommendation for Cryptographic Key Generation*, Revision 2, July 2019
[180]	NIST, *Secure Hash Standard*, FIPS Publication 180-4, August, 2015
[186]	NIST, *Digital Signature Standard (DSS)*, FIPS Publication 186-4, July, 2013
[197]	NIST, *Advanced Encryption Standard (AES)*, FIPS Publication 197, November 26, 2001
[198]	NIST, *The Keyed-Hash Message Authentication Code (HMAC)*, FIPS Publication 198-1, July 2008

Other References	
[APDUSpec]	*SE050 APDUSpecification, API description SE050 IoT Applet,* NXP Semiconductors, Rev 2.8, 19 December 2019. https://www.nxp.com/docs/en/application-note/AN12413-SE050_APDU_specification.pdf
[DTR]	NIST, *Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules*, January 2011
[FastSCP]	*'FastSCP' Secure Channel Protocol*, NXP Semiconductors, Rev 1.0, 3 April 2015
[FIPS140-2]	NIST, *Security Requirements for Cryptographic Modules*, May 25, 2001

Other References	
[GlobalPlatform]	*GlobalPlatform Card Specification 2.3, GlobalPlatform Inc., December 2015* *GlobalPlatform Consortium: GlobalPlatform Card -- Confidential Card Content Management -- Card Specification 2.2 -- Amendment A*, January 2011 *GlobalPlatform Consortium: GlobalPlatform Card Technology -- Contactless Services -- Card Specification v2.2 -- Amendment C*, July 2014
[IG]	NIST, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program,* last updated December 3, 2019
[ISO 7816]	ISO/IEC 7816-1: 2011 *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/IEC 7816-6:2016 *Identification cards -- Integrated circuit cards -- Part 6: Interindustry data elements for interchange* ISO/IEC 7816-8:2016 *Identification cards -- Integrated circuit cards – Part 8: Commands and mechanisms for security operations* ISO/IEC 7816-12:2005 *Identification cards -- Integrated circuit cards -- Part 12: Cards with contacts -- USB electrical interface and operating procedures* ISO/IEC 7816-15:2016 *Identification cards – Integrated circuit cards – Part 15: Cryptographic Information application*
[ISO 14443]	ISO/IEC 14443-3:2016 *Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision* ISO/IEC 14443-4:2016 *Identification cards -- Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol*
[JavaCard]	*Java Card 3.0.5 Runtime Environment (JCRE) Specification*, *May 2015* *Java Card 3.0.5 Virtual Machine (JCVM) Specification*, *May 2015* *Java Card 3.0.5 Application Programming Interface* Published by Oracle
[NXP T1I2C]	*NXP T=1 Over SPI/I2C Specification*, January 9 2019
[PKCS#1]	*PKCS #1 v2.1: RSA Cryptography Standard*, RSA Laboratories, June 14, 2002
[RFC5639]	Request for Comments: 5639, *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*, March 2010
[SCP03]	*GlobalPlatform Card Technology, Secure Channel Protocol 03, Card Specification v 2.2 – Amendment D,* Global Platform, Version 1.1.1
[SEC2]	*SEC 2: Recommended Elliptic Curve Domain Parameters,* Certicom Research, January 27, 2010 Version 2.0
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, Revision 2, March 2019

Acronyms and Definitions

Table 2: Acronyms and Definitions

Acronym	Definition
APDU	Application Protocol Data Unit, see [ISO 7816]
API	Application Programming Interface
BCD	Binary-Coded Digit
CM	Card Manager, see [GlobalPlatform]
CRNGT	Continuous random Number Generator Test, see AS09.42
CSP	Critical Security Parameter, see [FIPS 140-2]
DAP	Data Authentication Pattern, see [GlobalPlatform]
DPA	Differential Power Analysis
GP	GlobalPlatform
HID	Human Interface Device (Microsoftism)
IC	Integrated Circuit
ISD	Issuer Security Domain
I^2C or I2C	Inter-Integrated Circuit, see [NXP T1I2C]
ISD	Issuer Security Domain, see [GlobalPlatform]
KAT	Known Answer Test
NVM	Non-Volatile Memory (e.g., EEPROM, Flash)
OP	Open Platform (predecessor to GlobalPlatform)
PCR	Platform Configuration Register
PCT	Pairwise Consistency Test
PKI	Public Key Infrastructure
SCP	Secure Channel Protocol, see [GlobalPlatform]
SSD	Secondary Security Domain
SPA	Simple Power Analysis
TPDU	Transaction Protocol Data Unit, see [ISO 7816]

1 Overview

This document defines the Security Policy for the NXP Semiconductors SE050 cryptographic module, hereafter denoted *the Module* or *Secure Element*. The Module, validated to FIPS 140-2 overall Level 3, is a single chip module named SE050 implementing the GlobalPlatform operational environment (Card Manager (ISD/SSD)) and an application, the SE050 IoT applet v3.6.0.

The Module is a non-modifiable operational environment under the FIPS 140-2 definitions. New firmware cannot be loaded into this Module.

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

Table 3: Security Level of Security Requirements

Security Requirement	Level
Cryptographic Module Specification	3
Cryptographic Module Ports and Interfaces	3
Roles, Services, and Authentication	3
Finite State Model	3
Physical Security	4
Operational Environment	N/A
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

1.1 Versions, Configurations and Modes of Operation

The SE050 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 SE050 Card Manager component.

Part number	Interface	Hardware Version	Platform ID	ROM ID	Patch ID
SE050	Dual + I2C	N7121 B1	4A335233353130323634353731313030034D67740BE14219	2E5AD88409C9BADB	1

Table 4: Operating system identification

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

Code	Value	Parameter settings
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

Table 5: APDU command

The command answers the content of the DF28 file. The platform version is located at the tag ‘03’, the value is 4A335233353130323634353731313030034D67740BE14219.
Tag 02 identifies the patch version: 0000000000000001.
Tag 08 identified the ROM ID: 2E5AD88409C9BADB.

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 FIPS compliancy, where ‘00’ identified FIPS mode not active and ‘01’ - FIPS mode active.

The SE050 IoT applet v3.6.0 of the Module is configured to always runs in an Approved mode of operation.

The personalized product shall have the applet identification:

- Package ID: A000000396545300000000103000200H
- Applet ID: A000000396545300000000103000000000H
- Instance ID: A000000396545300000000103000000000H

To verify that the SE050 IoT applet v3.6.0 mode of operation, the SELECT APDU command (*Context* service) will be called with the following parameters: CLA = 00, INS = A4, P1 = 04, P2 = 00, Lc = 10, Incoming Data = A000000396545300000000103000000000, and Le = 00.

The Module shall answer 03060061D2010B followed by status code 9000. The response includes the BCD encoded applet version (030600) and the supported applet feature bitmap (61D2). It is not possible in any way to modify the applet version or the supported features bitmap after the device leaves the factory.

The tested configuration of the product identified here has exactly one applet instance: the SE050 IoT Applet instance. No other applet instance is allowed.

1.2 Hardware and Physical Cryptographic Boundary

The Module is designed to be used as a part of an IoT system. It works as an auxiliary security device attached to a host controller. The physical form of the Module is depicted in Figure 1 (to scale); the red outline depicts the physical cryptographic boundary, representing the surface of the chip and the bond pads.
In production use, the Module is delivered to either vendors or end user customers in a HX2QFN20 (SOT1969-1) package, the package is excluded from the FIPS140-2 security testing. The package dimensions are 3 mm x 3 mm x 0.32 mm with a 0.4 mm pitch.

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 T1I2C] connection to a host controller.

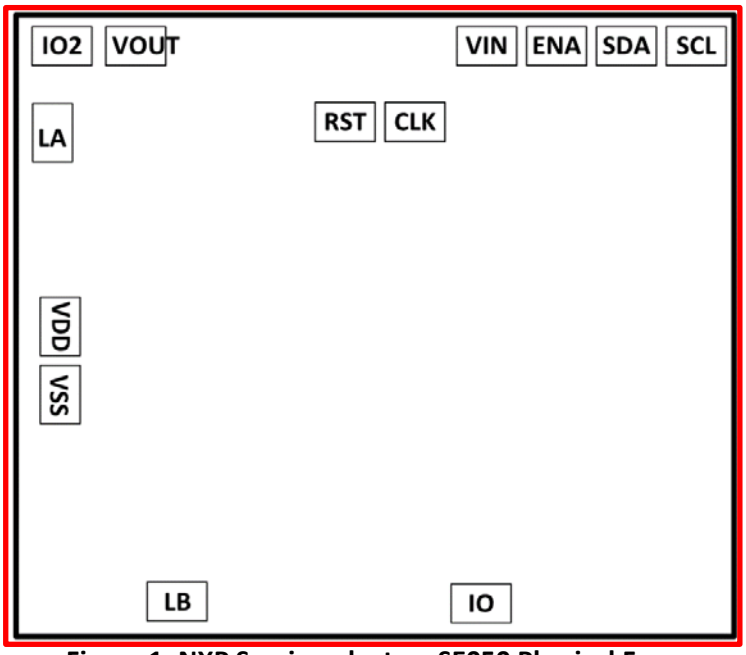

Figure 1: NXP Semiconductors SE050 Physical Form

Port	Communication Mode	Description	Logical Interface Type
VSS, VDD	Contact	Supply voltage	Power
VIN, VOUT	Contact, Contactless, I2C	Supply voltage and logic supply in case deep power-down mode is used.	Power
ENA	N/A	Deep power-down mode enabled	Control in
RST	Contact	Reset	Control in
CLK	Contact	Clock	Control in
IO	Contact	Input/Output	Control in, Data in, Data out, Status out
	I2C	Master SDA	
IO2	Contact	Input/Output 2	Control in, Data in, Data out, Status out
	I2C	Master SCL	
LA, LB	Contactless	Antenna	Power, Control in, Data in, Data out, Status out
SDA	I2C	Slave data	Control in, Data in, Data out, Status out
SCL	I2C	clock	Control in

Table 6: Ports and Interfaces

1.3 Firmware and Logical Cryptographic Boundary

Figure 2 depicts the Module operational environment.

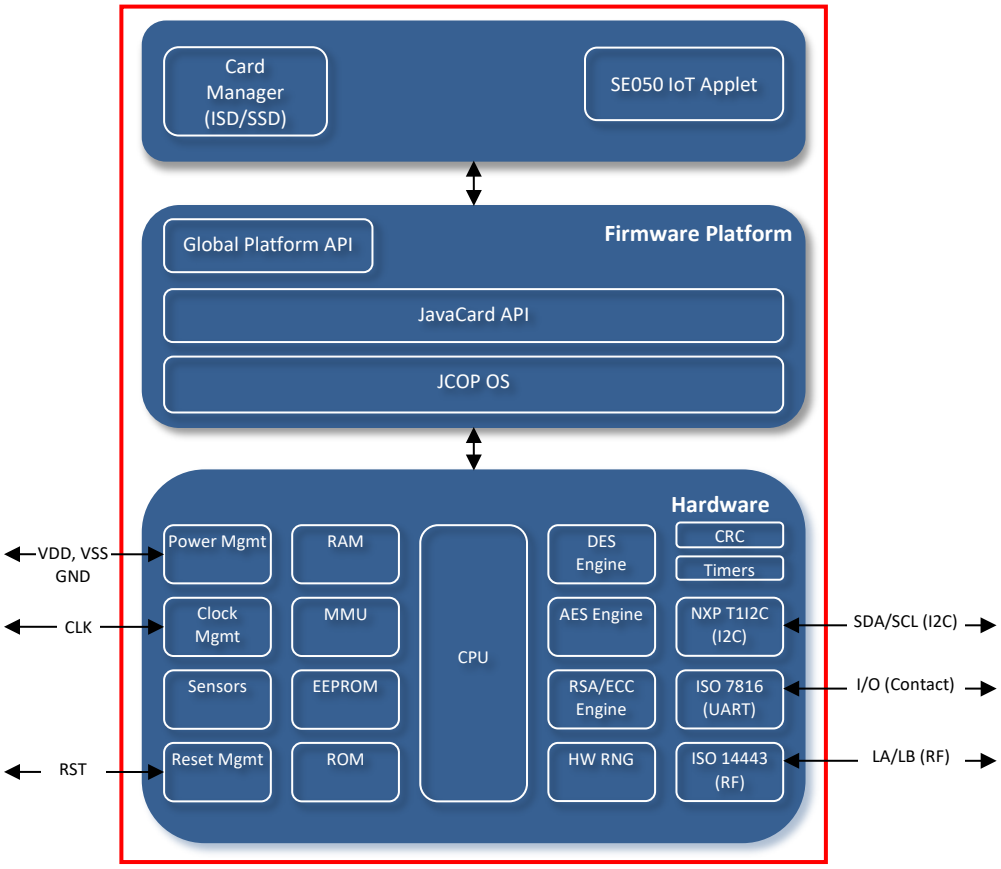

Figure 2: Module Block Diagram

The JavaCard and Global Platform APIs are internal interfaces available to applets. Only IoT applet and Card Manager (ISD/SSD) services are available at the card edge (the interfaces that cross the cryptographic boundary).

The product is delivered with the SE050 IoT applet installed and configured before product's delivery to customer. The end-user can personalize the Module with its objects but cannot modify the configuration of the Module, the Module always operates in an Approved mode of operation.

2 Cryptographic Functionality

The Module implements the Approved and Allowed cryptographic functions[1] listed below.

CAVP Cert	Algorithm	Standard	Mode/ Method	Description	Use
C880	AES	[197], [38A]	CBC, ECB, CTR	AES-128, AES-192, AES-256	Data Encryption/ Decryption
Vendor Affirmed	AES CBC CS	[197], [38ACS]	CBC-CS3	AES-128	Data protection
C880	AES CMAC	[197], [38B]	CMAC	AES-128, AES-192, AES-256	Message Authentication; generation and verification SP 800-108 KDF
Vendor Affirmed	CKG	[133]	§4: Using the Output of a Random Bit Generator		Vendor Affirmed, Asymmetric Key Generation is based on unmodified output of the DRBG cert. #C886
C1429	CVL	[56A]	ECC CDH Primitive	P-256	Shared Secret Computation
C838	CVL	[56B]	RSADP	n=2048, 3072, 4096	RSA encryption primitive. 3072 and 4096-bit are approved per IG A.14.
C838	CVL	[56B]	RSADP	n=2048, 3072, 4096	RSA decryption primitive based on RSA CRT. 3072 and 4096-bit are approved per IG A.14.
C886	DRBG	[90A]	CTR_DRBG	AES-128, AES-256	Deterministic Random Bit Generation AES-128: RSA key generation AES-256: ECDSA key generation
C1429	ECDSA	[186]	P-224, P-256, P-384, P-521		ECC Key Generation
			P-224: (SHA-224, SHA-256, SHA-384, SHA-512), P-256: (SHA-256, SHA-384, SHA-512), P-384: (SHA-384, SHA-512), P-521: (SHA-512)		Digital Signature Generation
			P-224: (SHA-224, SHA-256, SHA-384, SHA-512), P-256: (SHA-256, SHA-384, SHA-512), P-384: (SHA-384, SHA-512), P-521: (SHA-512)		Digital Signature Verification
C1426	HMAC	[198]	SHA-1, SHA-256, SHA-384, SHA-512		Message Authentication, key strength > 112 bits

[1] Some cryptographic implementations have been tested with additional modes of operation or key sizes but only the modes of operation and key sizes of the cryptographic algorithms listed in the table are used by the module.

Vendor Affirmed	KAS-SSC	[56Arev3]	OnePass EC Diffie-Hellman	P-256	FastSCP shared secret computation
C1428	KBKDF	[108]	Counter	AES-128, AES-192, AES-256	Deriving keys from existing keys
Vendor Affirmed	KDA	[56Crev1]	One-step key-derivation functions option 1	SHA-256	FastSCP session key derivation
C880	KTS	[38F]	AES CBC / AES CMAC	AES-128, AES-192, AES-256	Meets the SP 800-38F §3.1 ¶3 requirements for symmetric key wrapping, using Cert. #C880 AES and AES CMAC. Key establishment methodology provides between 128 and 256 bits of encryption strength.
C1427	RSA	[186]	n=2048, 3072		Key Generation
C838	RSA	[186]	n=2048, 3072, 4096 with PKCS v1.5 and PKCSPSS and SHA-(224, 256, 384, 512)		Digital Signature Generation 4096-bit RSA Signature Generation was tested against FIPS 186-2, allowed per IG G.18
			n=2048, 3072, 4096 with PKCS v1.5 and PKCSPSS and SHA-(1[2], 224, 256, 384, 512)		Digital Signature Verification 4096-bit RSA Signature Verification is approved per IG A.14
C837	SHS	[180]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		Message Digest Generation
C880	Triple-DES[3]	[67]	CBC, ECB	3-Key	Data Encryption and Decryption

Table 7: Approved Algorithms

Algorithm	Description
NDRNG	Hardware RNG; used as entropy input to the FIPS approved (Cert. #C886) DRBG. The non-deterministic RNG provides a minimum entropy of 128 bits for AES-128 CTR_DRBG and 256 bits for AES-256 CTR_DRBG.

Table 8: Non-Approved but Allowed Cryptographic Functions

EC	Standard	Strength	Singular	Field	Co-Factor
Brainpool224r1	[RFC5639]	112	No	IF_p	1
Brainpool256r1	[RFC5639]	128	No	IF_p	1
Brainpool320r1	[RFC5639]	128	No	IF_p	1

[2] This algorithm is Approved for legacy use

[3] The same Triple-DES key is not used more than either 2^{16} per IG A.13. When the block limit is reached the key value is cleared and the key is set to un-initialized automatically.

EC	Standard	Strength	Singular	Field	Co-Factor
Brainpool384r1	[RFC5639]	192	No	IF_p	1
Brainpool512r1	[RFC5639]	256	No	IF_p	1
Secp224k1	[SEC2]	112	No	IF_p	1
Secp256k1	[SEC2]	128	No	IF_p	1

Table 9: Non-Approved but Allowed Elliptic Curves used with ECDSA

2.1 Critical Security Parameters and Public Keys

All CSPs used by the Module are described in this section. All usage of these CSPs is described in the services detailed in Section 3.3. In the tables below, the following prefixes are used:

- OS prefix denotes operating system.
- SD prefix denotes a GlobalPlatform Security Domain.
- APP prefix denotes an Applet CSP or a Public Key.

CSP	Description/Usage
	Card Manager/SSD
OS-DRBG-EI	NDRNG entropy input to CTR_DRBG.
OS-DRBG-STATE	880-bit value; the current DRBG state.
OS-SKEK	128-bit key stored in NVM, used to derive OS-MKEK.
OS-MKEK	AES-128 key used to encrypt all secret and private key data stored in NVM.
SD-KENC	AES (256-bit) Master key used to derive SD-SENC.
SD-KMAC	AES (256-bit) Master key used to derive SD-SMAC.
SD-KDEK	AES (256-bit) Sensitive data decryption key used to decrypt CSPs.
SD-SENC	AES (256-bit) Session encryption key used to encrypt / decrypt secure channel data.
SD-SMAC	AES (256-bit) Session MAC key used to verify inbound secure channel data integrity.
SD-RMAC	AES (256-bit) Session MAC key used to generate response secure channel data MAC.
	IoT applet
APP-TRANSPORT-CIPHER	256-bit AES-CBC encryption key used to either export or import keys or data.
APP-TRANSPORT-MAC	128-bit AES-CMAC authentication key used to either export or import another key.
APP-KAS-SSC-EC-PRIV-KEY	P-256 KAS Shared Secret computation private key.
APP-KAS-SS	KAS Shared Secret CSP.
APP-AES-KEY-AUTH	128-bit AES key used in AESKey session or ECKey session authentication methods.
APP-SENC	AES 128-bit AESKey or ECKey session encryption key used to encrypt / decrypt secure channel data.
APP-SMAC	AES 128-bit AESKey or ECKey session MAC key used to verify inbound secure channel data integrity.

CSP	Description/Usage
APP-RMAC	AES 128-bit AESKey or ECKey session MAC key used to generate response secure channel data MAC.
APP-USERID-FILE	4 to 16-byte UserID authentication data.
APP-EC-PRIV-KEY	Elliptic curve private key that allows to perform ECDSA cryptographic operations, using NIST P-224, P-256, P-384 or P-521, Brainpool 224, 256, 320, 384 or 512-bit curves, secp224k1 or secp256k1 curve.
APP-RSA-PRIV-KEY	2048-bit, 3072-bit or 4096-bit RSA private key that allows to perform RSA cryptographic operations.
APP-AES-KEY	AES (128, 192 or 256 bits) key used to perform AES cipher mode operations.
APP-DES-KEY	3-key Triple-DES key used to perform Triple-DES cipher mode operations.
APP-HMAC-KEY	(112-bit and above) HMAC keys used to perform KDF or HMAC operations.

Table 10: Critical Security Parameters

Public Key	Description/Usage
	Applet
APP-KAS-SSC-EC-PUB-KEY	P-256 KAS Shared Secret computation public key.
APP-EC-PUB-KEY-CO	P-256 ECDSA public key used to authenticate the CO.
APP-EC-PUB-KEY-USER	P-256 ECDSA public key used to authenticate as user.
APP-EC-PUB-KEY	Elliptic curve public key that allows to execute EC cryptographic operations (keys can be inserted by users).
APP-RSA-PUB-KEY	RSA public key that allows to execute RSA cryptographic operations (keys can be inserted by users).

Table 11: Public Keys

3 Roles, Authentication and Services

The Module:

- Does not support a maintenance role.
- Clears previous authentications on power cycle.
- Supports concurrent operators

Table 12 lists all operator roles supported by the Module.

Role ID	Role Description
CO	Cryptographic Officer – manages Module content and configuration, including management of Module data via the SSD. Authenticated as described in *Platform authentication* and *Applet authentication (ECKey session)* in sub-section below.
User	The device Holder (applet user) – performs FIPS approved cryptographic operations. Authenticated as described in *Platform authentication* and *Applet authentication* in sub-section below.

Table 12: Roles Supported by the Module

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.
- CO authentication method does not exchange plaintext CSP.
- User authentication data are encrypted and authenticated during entry with GlobalPlatform SCP03, are stored encrypted with OS-MKEK and is only accessible by authenticated services.

3.1 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, 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.
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 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 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).

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

- 1/(2^128) = 2.9E-39 (MAC||cryptogram, using a 128-bit block for authentication)

This authentication method includes a counter of failed authentication called “velocity checking” by GlobalPlatform. The counter is decremented prior to any attempt to authenticate and is only reset to its threshold (maximum value) upon successful authentication.

The Module enforces a maximum of 60 failed Global Platform SCP03 authentication attempts before blocking permanently the card. The probability that a random attempt will succeed over a one-minute interval is:

- 60/(2^128) = 1.7E-37 (MAC||cryptogram, using a 128-bit block for authentication)

3.2 Applet authentication

The applet allows creating authenticated session using an Authentication Object which can be either UserID session, AESKey session, or ECKey session.

Authenticated session allows users to protect and safeguard their credentials against third party use as only the authenticated user has proper rights on the credentials. This is ensured by applying correct policies to the credentials. A policy binds functional access to an Authentication Object where an Authentication Object represents a user. See sections 3.2.3 and 3.7 of [APDUSpec] for more details.

The different authentication methods are described in the sub-sections below.

3.2.1 UserID session

An UserID session authentication method is provided by the *Session management* service.
During a UserID session, the session user identifier (UserID) is verified in order to allow setting up a session. If the UserID is correct, the session establishment will succeed; otherwise, the session will not be opened.

An UserID can be configured from a minimum of 4 up to a maximum of 16 bytes (128 bits). In the worst-case scenario, a 4-byte UserID is used, the probability that a random attempt will succeed using this authentication method is:

- 1/(2^32) = 4.3E-9

The number of authentication attempts is configurable. It can be an infinite attempt number, or it can be limited by a counter comprised between 1 and 255 attempts. A maximum of 4700 authentications can be performed in one minute. In the worst-case scenario, the probability that a random attempt will succeed over a one-minute period is:

- 4700/(2^32) = 1.0E-6

3.2.2 AESKey session

The AESKey session authentication method is provided by the *Session management* service. The APP-AES-KEY-AUTH key is used to derive the APP-SENC, APP-SMAC keys, and APP-RMAC. 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.
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 64-bit cryptogram with APP-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 APP-SMAC key, the MAC is concatenated to the command, and the command is sent to the Secure Element. The Secure Element checks the message authentication code and compares the received cryptogram to the calculated cryptogram. If all of this succeeds, the two participants are mutually authenticated (the external entity is authenticated to the Module in the CO role).

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

- 1/(2^128) = 2.9E-39 (MAC||cryptogram, using a 128-bit block for authentication)

The number of authentication attempts is configurable. It can be an infinite attempt numbers or it can be limited by a counter comprised between 1 and 32767. A maximum of 4700 authentications can be performed in one minute. In the worst-case scenario, the probability that a random attempt will succeed over a one-minute period is:

- 4700/(2^128) = 1.3E-35 (MAC||cryptogram, using a 128-bit block for authentication).

3.2.3 ECKey session

An ECKey session authentication method is provided by the Session management service.

The ECKey session authentication method consist in verifying a P-256 ECDSA signature. The P-256 EC public key is either initially imported by the User (APP-EC-PUBLIC-KEY-USER) or provisioned during the manufacturing (APP-EC-PUBLIC-KEY-CO). The user will own the corresponding ECDSA private key.
In addition to User's authentication, ECKey session is used to establish APP-KAS-SS with the Approved KAS algorithm. The shared secret is used to derive the AES-128 APP-AES-KEY-AUTH which is itself used to derive the APP-SENC, APP-SMAC and APP-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.
First, the user requests the Module public key APP-KAS-SSC-EC-PUB-KEY, this key is signed with the private key APP-KAS-SSC-EC-PRIV-KEY by the Module. Then, the User sends the ephemeral KAS public key signed with User's ECDSA private key. Finally, the Module verifies the ECDSA signature of the ephemeral key with either APP-EC-PUBLIC-KEY-USER or APP-EC-PUBLIC-KEY-CO before to initiate the KAS shared secret computation.

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

- 1/(2^256) = 8.6E-78 (using a 256-bit EC key for authentication)

The number of authentication attempts is configurable. It can be an infinite attempt numbers or it can be limited by a counter comprised between 1 and 32767. A maximum of 4700 authentications can be performed in one minute. In the worst-case scenario, the probability that a random attempt will succeed over a one-minute period is:

- 4700/(2^256) = 4.0E-74 (using a 256-bit EC key for authentication)

3.3 Services

All services implemented by the Module are listed in the tables below. The ISD / SSD Services are provided by the Card Manager. Such services can be accessed directly with a selection of Security Domain or through the SE050 IoT applet for the SSD. The Applet Services are provided by the SE050 IoT applet.

Service	Description
	ISD / SSD Services
Card Reset	Power cycle or reset the Module.
Context	Select an applet or manage logical channels.
Info	Read unprivileged data objects, e.g., Module configuration or status information (Show Status). This service includes the Power-On Self-Test on-demand.

Table 13: Unauthenticated Services

Service	Description	CO	User
	ISD / SSD Services		
Lifecycle	Modify the card or applet life cycle status.	X	
Manage Content	Load keys and data.	X	
Privileged Info	Read Module data (privileged data objects, but no CSPs).	X	
Secure Channel	Establish and use a secure communications channel.	X	
	Applet Services		
Service			
Module Management	This service manages the SE050 applet.	X	
Session Management	This service manages the applet sessions. Users can decide to open a session or not. Opening a session requires to authenticate to the applet using either an UserID, an AES128 key or an EC key depending on the session type.	X	X
Secure Object Write Functionality	This service manages the generation (either an RSA or EC key pair) or transport (EC keys, RSA keys, symmetric keys, binary files, UserIDs, monotonic counters, PCRs) of Secure Objects.	X	X
Secure Object Read Functionality	This service manages the reading of Secure Objects or its attributes. Asymmetric private keys or symmetric keys can never be read in plaintext.	X	X
Secure Object Management	This service manages the reading of Secure Object attributes.	X	X
EC Curve Management	This service manages the EC curves that can be used during EC cryptographic operations.	X	X
Crypto Object Management	This service manages the Crypto Objects that can be used. Crypto Objects allow to do operations in multiple steps (init/update/final). Supported Crypto Objects allow to use a digest, cipher or MAC algorithm to be used.	X	X
EC Crypto Operations	This service triggers OS API for ECDSA signature generation and verification and for EC DH shared secret calculation according [56A-rev3] §5.7.1.2.	X	X
RSA Crypto Operations	This service triggers OS API for RSA signature generation and verification and for RSA encryption and decryption.	X	X
Symmetric Cipher Crypto Operations	This service triggers OS API for AES and Triple-DES encryption and decryption.	X	X
MAC Calculation Crypto Operations	This service triggers OS API for MAC Calculation.	X	X
Secure Hash Crypto Operations	This service triggers OS API for [FIPS 180-4] compliant hash algorithms.	X	X

Table 14: Authenticated Services

Table 15 below describes the access to CSPs by service with brief descriptions, which are intended to help readers understand the patterns of access. Explanations are provided in groups of services and/or keys (as best suited to explain the pattern of access), describing those aspects that have commonality across services or keys/CSPs.

Lifecycle: must be used with Secure Channel active (hence SD Session keys are 'E'); zeroizes all keys except session keys when Lifecycle is used for card termination.

OS-SKEK: generated on first power-up of the Module in a manufacturing setting; used to derive OS-MKEK; zeroized on Lifecycle card termination.

OS-MKEK: derived from OS-SKEK; used whenever any private or secret key is accessed; zeroized on Lifecycle card termination.

OS-DRBG CSPs: OS-DRBG-EI is the NDRNG entropy input to the DRBG instantiation at power-on (Module Reset), zeroized after use. OS-DRBG-STATE is generated at startup (Module Reset), zeroized at shutdown as part of Module Reset, or by LifeCycle card termination. Each 'E' in the OS-DRBG-STATE column indicates the use of the DRBG to generate keys (or nonces), as the value is used and the state is updated.

Secure Channel Master Keys (SD-KENC, SD-KMAC): 'E' when a secure channel is initialized (GP Secure Channel). May be updated ('I') using the Manage Content service; zeroized by Lifecycle card termination.

SD-KDEK: is used to decrypt CSPs entered into the module during the applet personalization.

Secure Channel Session Keys (SD-SENC, SD-SMAC, SD-RMAC): 'E' for any service that are used with secure channel active. 'GE' on GP Secure Channel as a consequence of secure channel initialization and usage. 'Z' on Module Reset is a consequence of RAM clearing/garbage collection.

Applet CSPs (APP-): Applet CSPs and public keys are separated between cryptographic operations services and management services.

CSPs APP-EC-KEY, APP-RSA-KEY, APP-DES-KEY, APP-HMAC-KEY are called by the cryptographic operation services.

All other keys are either used to protect in confidentiality and to authenticate the data exchanged between an external entity and the Module, to authenticate the users, or to establish CSPs and public keys.

The transport mechanism allows exporting transient CSP keys stored on the module ('O' of *Secure Object Read Functionality* service for CSPs) and importing these exported keys only. The keys are protected during the transport with APP-TRANSPORT-CIPHER and APP-TRANSPORT-MAC over the Secure Channel. The public key can also be output 'O' in plaintext with the *Secure Object Read Functionality* service.

APP-KAS-SSC-PRIV-KEY and APP-KAS-SSC-PUB-KEY are used to compute the KAS shared secret PP-KAS-SS.

APP-AES-KEY-AUTH is the (master) key type for AESKey session authentication or establish with the ECCKey session KAS.

APP-SENC, APP-SMAC and APP-RMAC are the session keys used by the secure messaging in AESKey or ECKey sessions.

ECKey session authentication used either APP-EC-PUBLIC-KEY-CO or APP-EC-PUBLIC-KEY-USER.

Secure Object Management service can zeroize 'Z' all persistent object of the Module.

Secure Object Write Functionality service is used to either import CSPs and Public keys or to generate the asymmetric keys.

Services	CSPs																							
	OS-DRBG-EI	OS-DRBG-STATE	OS-SKEK	OS-MKEK	SD-KENC	SD-KMAC	SD-KDEK	SD-SENC	SD-SMAC	SD-RMAC	APP-TRANSPORT-CIPHER	APP-TRANSPORT-MAC	APP-KAS-SSC-EC-PRIV-KEY	APP-KAS-SS	APP-AES-KEY-AUTH	APP-SENC	APP-SMAC	APP-RMAC	APP-USERID-FILE	APP-EC-PRIV-KEY	APP-RSA-PRIV-KEY	APP-AES-KEY	APP-DES-KEY	APP-HMAC-KEY
Unauthenticated Role	*Card Manager*										*Applet*													
Card Reset	G Z	G EZ	E	G	--	--	--	Z	Z	Z	--	--	--	Z	--	Z	Z	Z	--	--	--	--	--	--
Context	--	--	--	--	--	--	--	EZ	EZ	EZ	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Info	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
CO	*Card Manager*										*Applet*													
Lifecycle	Z	Z	Z	Z	Z	Z	Z	EZ	EZ	EZ	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Manage Content	Z	Z	IZ	E	IE	IE	IE	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Privileged Info	--	--	--	E	E	E	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Secure Channel	G Z	G EZ	--	E	E	E	--	G E	G E	G E	--	--	--	--	--	--	--	--	--	--	--	--	--	--
CO / User	*Card Manager*										*Applet*													
Module Management	--	--	--	--	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Session Management	G Z	G EZ	--	E	E	E	--	G E	G E	G E	--	--	E	G EZ	E	G E Z	G EZ	G EZ	E	E	--	E	--	--
Secure Object Write Functionality	G Z	G EZ	--	E	--	--	--	E	E	E	E	E	--	--	I	--	--	--	I	GI	GI	I	I	I
Secure Object Read Functionality	--	--	--	E	--	--	--	E	E	E	E	E	--	--	--	--	--	--	--	O	O	O	O	O
Secure Object Management	--	--	--	E	--	--	--	E	E	E	--	--	--	--	Z	--	--	--	Z	Z	Z	Z	Z	Z
EC Curve Management	--	--	--	--	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Crypto Object Management	--	--	--	E	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	E	E	E
EC Crypto Operations	G Z	G EZ	--	E	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	E	--	--	--	--
RSA Crypto Operations	G Z	G EZ	--	E	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	E	--	--	--
Symmetric Cipher Crypto Operations	--	--	--	E	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	E	E	--
MAC Calculation Crypto Operations	--	--	--	E	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	E
Secure Hash Crypto Operations	--	--	--	--	--	--	--	E	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Table 15: CSPs Access within Services

Services	Public Keys				
	APP- KAS-SSC-EC-PUB-KEY	APP-EC-PUB-KEY-CO	APP-EC-PUB-KEY-USER	APP-EC-PUB -KEY	APP-RSA-PUB-KEY
Unauthenticated Role	*Applet*				
Card Reset	--	--	--	--	--
Context	--	--	--	--	--
Info	--	--	--	--	--
CO	*Applet*				
Lifecycle	Z	Z	Z	Z	Z
Manage Content	--	--	--	--	--
Privileged Info	--	--	--	--	--
Secure Channel	--	--	--	--	--
CO / User	*Applet*				
Module Management	--	IE	IE	--	--
Session Management	E	E	E	--	--
Secure Object Write Functionality	--	I	I	GI	GI
Secure Object Read Functionality	O	O	O	O	O
Secure Object Management	--	--	Z	Z	Z
EC Curve Management	--	--	--	--	--
Crypto Object Management	--	--	--	--	--
EC Crypto Operations	--	--	--	E	--
RSA Crypto Operations	--	--	--	--	E
Symmetric Cipher Crypto Operations	--	--	--	--	--
MAC Calculation Crypto Operations	--	--	--	--	--
Secure Hash Crypto Operations	--	--	--	--	--

Table 16: Public Keys Access within Services

- G = Generate: The service generates or derives the CSP/Public Key.
- I = Input: The service inputs the CSP/Public Key.
- E = Execute: The Module executes using the CSP/Public Key.
- O = Output: The service outputs the CSP/Public Key. CSP are always protected with the approved KTS.
- Z = Zeroize: The Module zeroizes the CSP/Public Key. For the Context service, SD session keys are destroyed on applet deselect (channel closure).
- -- = Not accessed by the service.

4 Self-Test

4.1 Power-On Self-Tests

The module benefit of IG 9.11, on the first power-on or on demand, the Module performs self-tests described in Table 17 below. All self-tests must be completed successfully prior to any other use of cryptography by the Module. If one of the self-tests fails, the system is halted and will start again after a reset.

For successive power-on, the Firmware Integrity (Flash and ROM) check is performed on every reset.

Test Target	Description
AES	Performs separate encrypt and decrypt KATs using an AES-128 key in CBC mode.
CMAC	Performs an AES-CMAC KAT with AES-128.
DRBG	Performs a fixed input KAT and all SP 800-90A health test monitoring functions.
ECDSA	Performs ECDSA signature generation and verification KATs using the P-521 curve and SHA-256; this self-test is inclusive of the CVL ECC CDH self-test.
Firmware Integrity	32-bit CRC performed over all code located in Flash and ROM.
HMAC	Performs a HMAC KAT with SHA-256.
KBKDF	Performs a KBKDF KAT with AES-128.
RSA	Performs separate RSA signature generation and verification KATs using an RSA 2048-bit key and SHA-256; this self-test is inclusive of the CVL RSA DP self-test.
SHA-1	Performs a fixed input KAT.
SHA-256	Performs a fixed input KAT (inclusive of SHA-224, per IG 9.4).
SHA-512	Performs a fixed input KAT (inclusive of SHA-384, per IG 9.4).
Triple-DES	Performs encrypt and decrypt KATs using 3-Key Triple-DES in CBC mode.

Table 17: Power-On Self-Test

All the Power-On Self-Test 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.

4.2 Conditional Self-Tests

Test Target	Description
DRBG CRNGT	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.
Generate PCT	Pairwise consistency test (Sign/Verify) performed when an asymmetric key pair is generated for RSA or ECC. The conditional test is implemented at the applet level.
NDRNG CRNGT	CRNGT is implemented following IG 9.8 by performing RCT on raw data (amongst other continuously running tests).
Signature PCT	Pairwise consistency test performed when a signature is generated for RSA or ECDSA.

Table 18: Conditional Self-Tests

5 Physical Security Policy

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 and is protected by active shielding (a grid of top metal layer wires with tamper response). A tamper event detected by the active shield places the Module permanently into the *TAMPER* error state. The Module includes also Environmental Failure Protection features, see section 6 below.

The Module is intended to be mounted in additional packaging; physical inspection of the die is typically not practical after packaging.

6 Mitigation of Other Attacks Policy

The Module is protected against 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, and detection of illegal address or instruction. 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.

7 Security Rules and Guidance

The Module implementation also enforces the following security rules:

1. The Module provides two distinct operator roles: User and Cryptographic Officer.
2. The Module does not support a maintenance interface or role.
3. The Module provides identity-based authentication.
4. The Module clears previous authentications on power cycle.
5. Power up self-tests do not require any operator action.
6. The Module allows the operator to initiate self-tests on-demand.
7. Data output is inhibited during key generation, self-tests, zeroization, and error states.
8. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.
9. The Module does not enter or output plaintext CSPs.
10. There are no restrictions on which CSPs are zeroized by the zeroization services, see Table 15.
11. The Module does not support manual key entry.
12. The Module does not output intermediate key values.
13. The module does not provide bypass services or ports/interfaces.
14. No additional interface or service is implemented by the Module which would provide access to CSPs.

In addition, the following guidance shall be followed:

15. The default SSD CSPs SD-SENC, SD-SMAC and SD-RMAC must be changed at the reception of the Module.