Sansec HSM

Cryptographic Module

Hardware versions SecHSM V2-1 (AC) and SecHSM V2-1 (DC)

Firmware version v3.02.0025

FIPS 140-2 Non-Proprietary Security Policy

Version 1.5

Last update: 2022-01-06

Table of Contents

1. Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for version SecHSM V2-1 (AC) and SecHSM V2-1 (DC) of the Sansec HSM Cryptographic Module. It contains the security rules under which the module must be operated and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 3 module.

The following sections describe the cryptographic module and how it conforms to the FIPS 140-2 specification in each of the required areas.

1.1. Module Overview

The Sansec Hardware Security Module (HSM) is a multi-chip standalone hardware cryptographic module that provides data encryption, data decryption, signature generation, signature verification, message digest, message authentication code (MAC), random number generation and key management services to business systems.

A business system host connects to the HSM through the network using the TCP/IP protocol. The host identifies and authenticates against the HSM with a user application ID and password. Once authentication succeeds, the host requests cryptographic services to the HSM, which processes the requests and sends back the result. In addition, users of the module can access the management functions by connecting to serial port of the HSM through the management terminal. HSM can also connect to an external VGA screen as its monitor to perform the administrative functions together with the USB keyboard. A typical application scenario is shown in Figure 1.

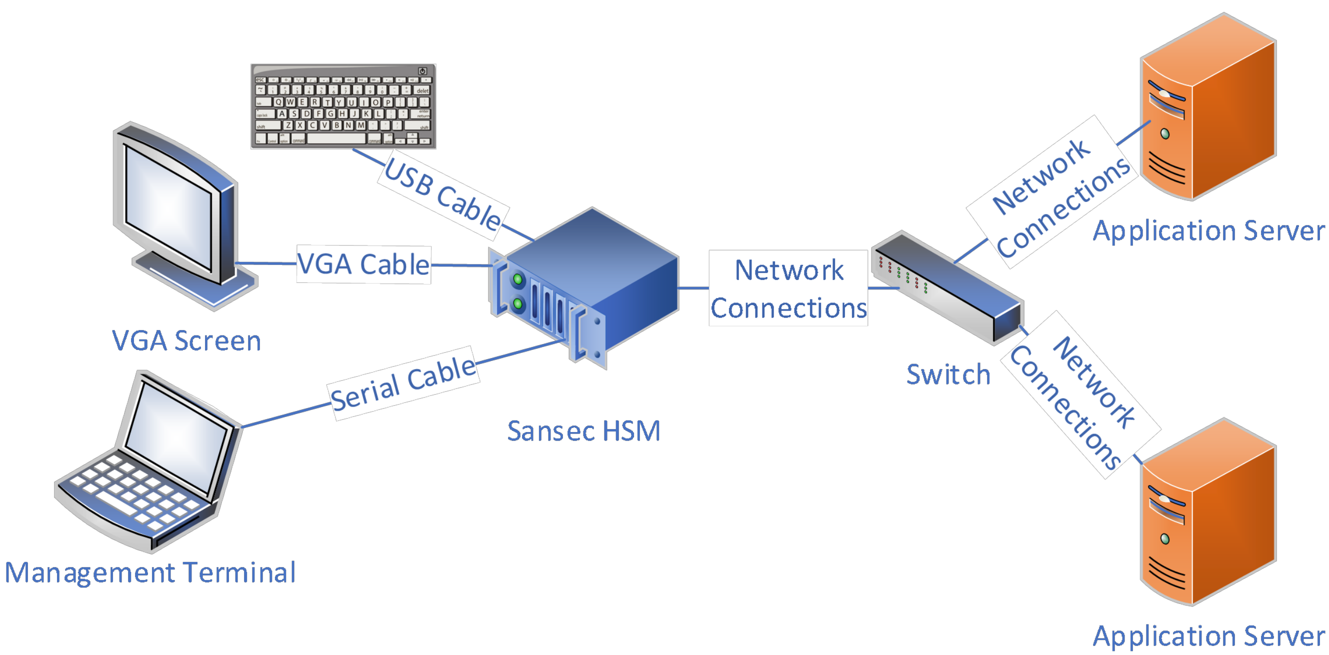

Figure 1 – Sansec HSM typical application scenario

The physical dimensions of the HSM are 447 mm x 86 mm x 500 mm (width x height x length), as shown in Figure 2.

Figure 2 – The Sansec HSM device and its physical external dimensions

The HSM provides a hardened, tamper-resistant environment. The HSM is enclosed entirely within an opaque secure steel chassis which deters physical tampering. The HSM also includes a tamper detection and response circuitry in the event the enclosure is ever opened.

The HSM module includes an Intel Celeron G4930 (Coffee Lake) processor. The module comes in two versions: model SecHSM V2-1 (AC) provides two AC power supply units (GW-CRPS550N), whereas model SecHSM V2-1 (DC) provides two DC power supply units (GW-CRPSD800).

1.2. Cryptographic Module Description

For the purpose of the FIPS 140-2 validation, the HSM is a multi-chip standalone hardware cryptographic module validated at an overall Security Level 3. The table below shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard:

FIPS 140-2 Section		Security Level
1	Cryptographic Module Specification	3
2	Cryptographic Module Ports and Interfaces	3
3	Roles, Services and Authentication	3
4	Finite State Model	3
5	Physical Security	3
6	Operational Environment	N/A
7	Cryptographic Key Management	3
8	EMI/EMC	3

FIPS 140-2 Section		Security Level
9	Self-Tests	3
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A
Overall Level		3

Table 1 – Security Levels

The cryptographic boundary of the module is defined as the entire HSM. The physical boundary of the cryptographic module is defined by the hard metal chassis, which surrounds all the hardware and firmware components of the module.

Figure 3 shows a hardware block diagram of the module. The blue bold line surrounding the hardware components represents the physical boundary of the module.

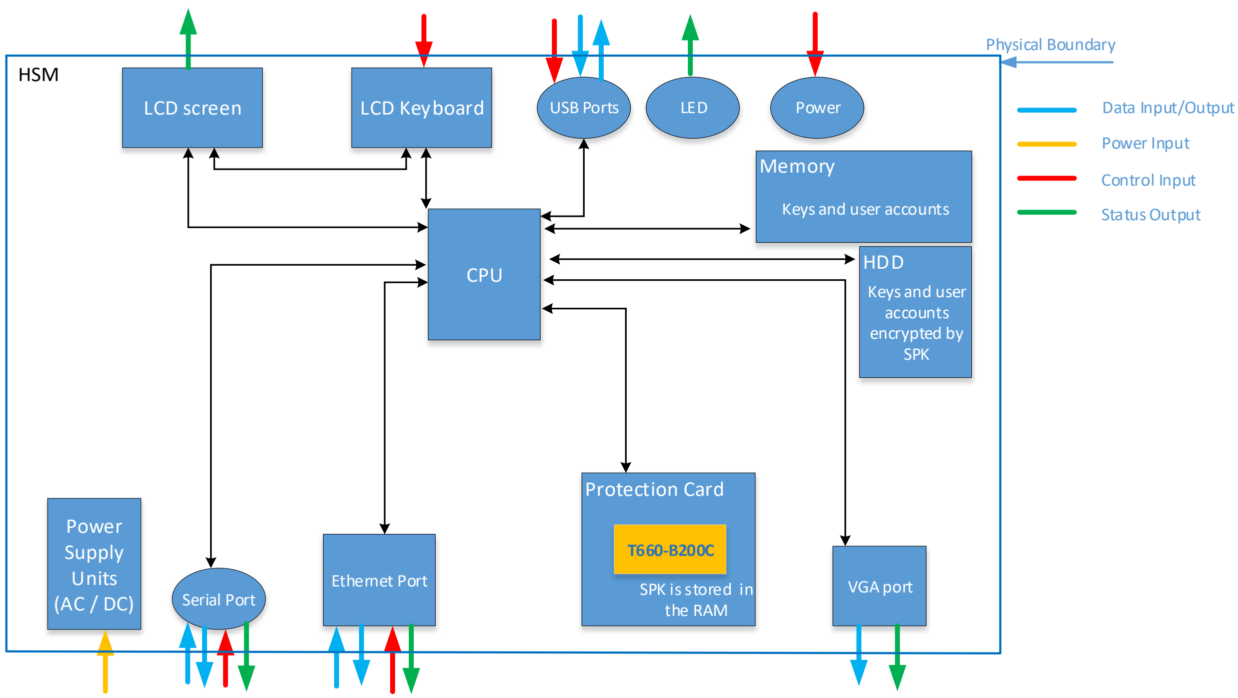

Figure 3 – Cryptographic Module Block Diagram

The HSM includes the Protection Card, a hardware component that is used for the identity-based authentication mechanism and the storage of user information and keys. The Protection Card implements some of the cryptographic algorithms provided by the HSM.

1.3. Mode of operation

The HSM supports two modes of operation.

- In "**FIPS mode**", (the Approved mode of operation) only approved or allowed security functions with sufficient security strength can be used.
- In "**non-FIPS mode**", (the non-Approved mode of operation) non-approved security functions can be used in addition to the security functions allowed in FIPS mode.

The mode of operation can be obtained as follows:

- By users of the module, using the “Show mode of operation” option in the management console (see Section 5.7 “Set FIPS mode” in [HSM-UM] for more information)
- By external entities (e.g. user applications) that connect through the network, using the service “Get FIPS / non-FIPS status” (command code “XA”), which returns the mode of operation (see Section 2.4.3 “Get Fips/No-Fips status (XA)” in [HSM-CS] for more information).

The FIPS mode of operation is implicitly assumed when a user initializes the HSM for the first time. Once the HSM is operational, a user with the Administrator role can set the mode of operation through the “Set FIPS mode” management service: the operator selects “1” for FIPS mode, or “2” for non-FIPS mode.

When a new mode of operation is selected by the user, the HSM erases all the information, and requires a new initialization. The cryptographic security parameters (CSP), which are encrypted in the module, are also erased. This mechanism ensures that CSPs used in FIPS mode of operation cannot be used in non-FIPS mode, and vice versa.

When the module is running in FIPS mode of operation, the module enforces that only service requests for approved cryptographic services, algorithms and key sizes are allowed.

2. Cryptographic Module Ports and Interfaces

2.1. Physical ports

Figure 1 shows the front panel of the HSM. The USB ports are used for authenticating users with credentials stored in their USB tokens. These ports are also used to input and output key components. The LCD screen provides a menu where options can be selected via the control buttons so module status information can be shown. The front panel also provides a power switch button to power on and off the HSM, and indicator lights for power, hard disk activity and network connections.

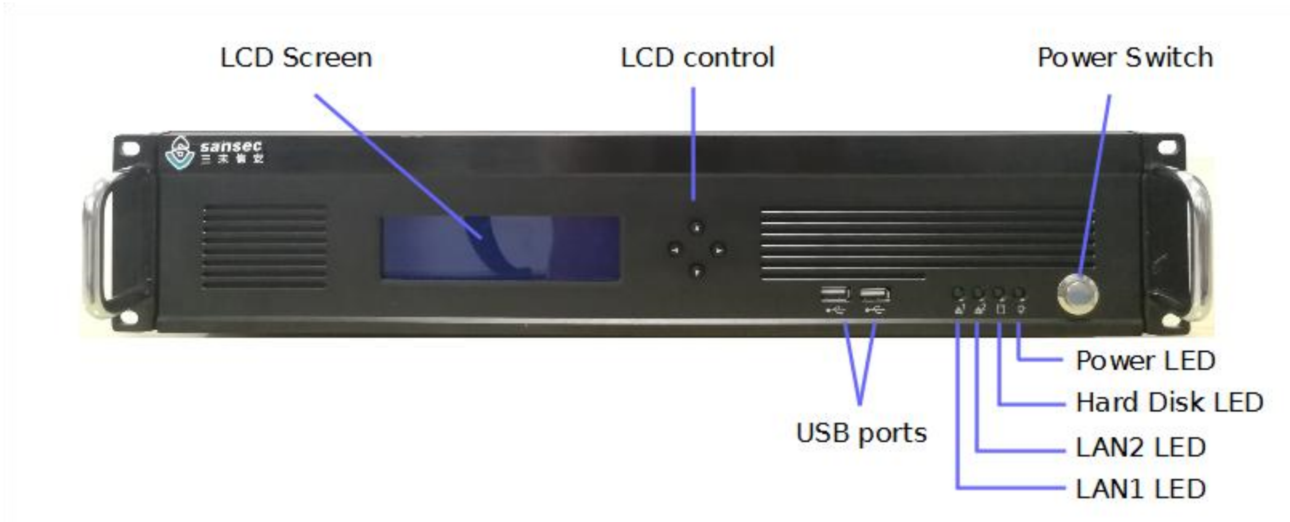

Figure 4 - Physical ports (front view)

Figure 5.a and Figure 5.b show the rear panel of the HSM. The HSM model SecHSM V2-1 (AC) includes two redundant AC power supply units, and SecHSM V2-1 (DC) includes two DC redundant power supply units, customer could choose different model according to their type of electric power line. The rear panel also includes a RS-232 port for connecting the serial console; and four RJ-45 jacks serve as Ethernet ports for connecting the HSM to the network. One IPMI LAN port which is disabled in factory and cannot be accessed by user. There are also four USB ports and one VGA port.

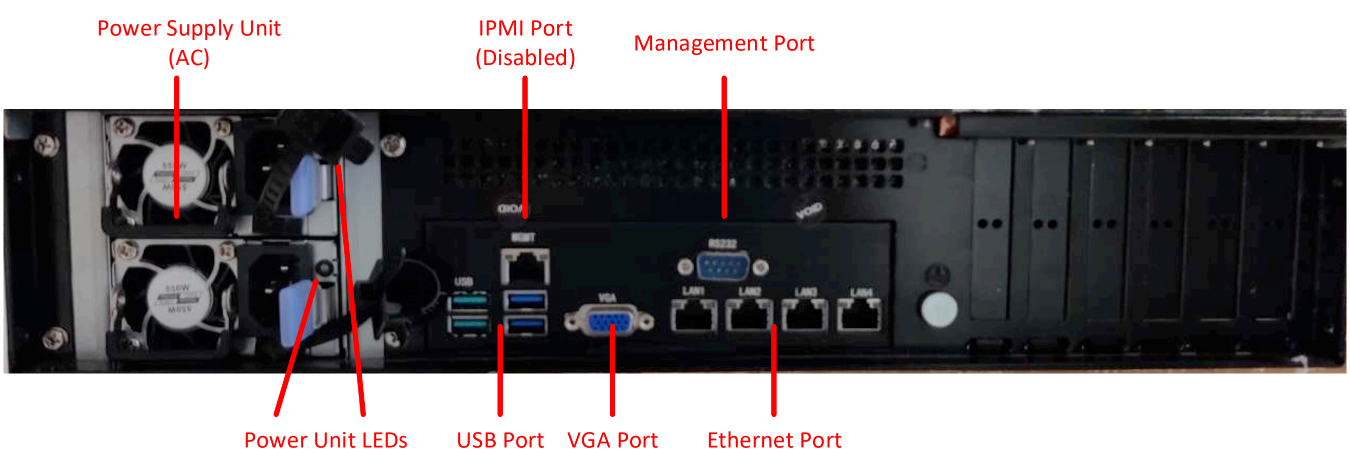

Figure 5.a - Physical ports SecHSM V2-1 (AC) *(rear view)*

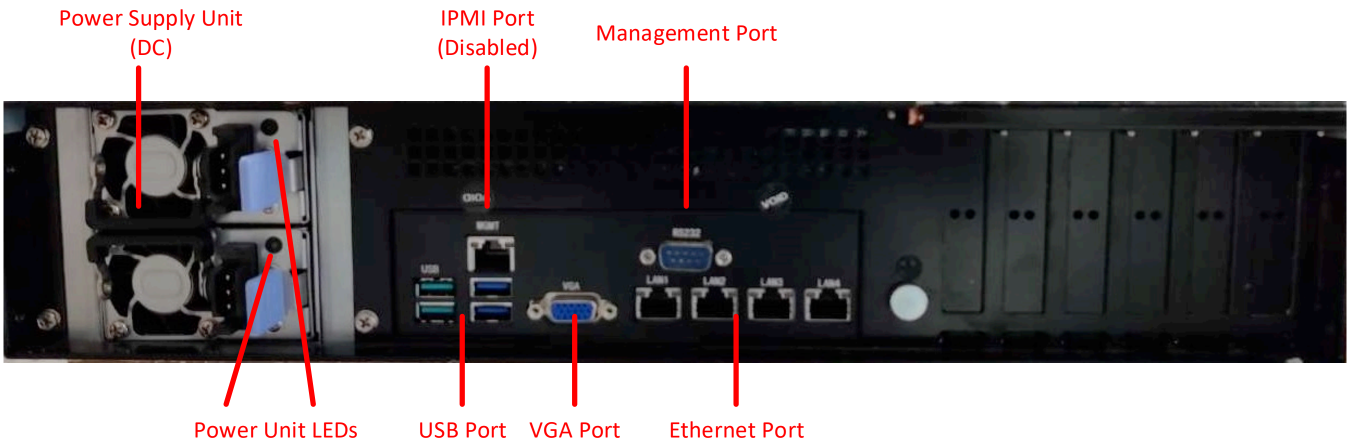

Figure 5.b - Physical ports SecHSM V2-1 (DC) *(rear view)*

The table below summarizes the physical ports.

Physical Ports		Description
1	USB ports (6)	Used for connecting USB tokens and keyboard
2	LCD screen	Displays module status information
3	LCD control buttons (4)	Control LCD screen menu
4	LED indicator lights (4)	Show activity in each network connection and hard disk, and power
5	Power inlet ports (2)	Redundant power supply units, the module could take either AC or DC power supply depending on the inserted units. AC and DC power supply units are equivalent.
6	Power unit LED (2)	LED indicator that is activated when the power unit is connected.
7	Power buzzer	Sound alarm that is activated when one of the power units is disconnected.
8	Ethernet ports (4)	RJ45 jacks that provide connection to the network
9	Management port	RS-232 interface used to connect a GPC to act as the HSM serial console
10	IPMI port	Disabled in factory and cannot be accessed by user
11	VGA port	Video connector

Table 2 – Physical ports of the module

2.2. Logical Interfaces

The following table summarizes the four logical interfaces and the mapping with the physical ports.

FIPS Interfaces	Physical Ports	Logical Interfaces
Data Input	USB ports	Data input fields in APDU messages.
	Management port	Data input through management console.
	Ethernet ports	Data input fields in service request messages. Key backup archive for restoring keys in module.
Data Output	USB ports	Data output fields in APDU messages.
	Management port	Data output shown by management console.
	Ethernet ports	Data output fields in service response messages. Key backup archive for safeguarding keys.
	VGA port	Data output shown by VGA screen.
Control Input	Power switch button	Power on or off the module.
	LCD screen buttons	Menu control and selection in LCD screen.
	Management port	Commands invoked in management console.
	Ethernet ports	Control input fields in service request messages.
	USB Ports	Commands input through keyboard.
Status Output	LCD screen	Data output.
	LED indicator lights	Indicate power, hard disk activity and network activity.
	Management port	Status output shown by management program.
	Ethernet ports	Status fields in service response messages.
	VGA port	Status output shown by VGA screen.
	Power unit LED	LED indicator that is activated when the power unit is connected.
	Power buzzer	Sound alarm that is activated when one of the power units is disconnected.
Power Input	Power units	Not applicable.

Table 3 – Logical Interfaces and their mapping with physical ports

3. Roles, Services and Authentication

This section defines the roles, services and authentication mechanisms with respect to the applicable FIPS 140-2 requirements.

The HSM implements identity-based authentication to authenticate the user and verify that the user is authorized to assume the assigned role. For those services that require identification and authentication, the HSM verifies that the user has the proper role to perform the service.

3.1. Roles

The HSM supports five roles: Super Administrator, Administrator, Operator, Auditor and User Application. The first four roles are crypto-officer roles and can be assigned to users of the module (user) that perform management operations. The User Application role is a user role assigned to external entities (user application) that connect to the module through the network port to request cryptographic services.

A user can have only one fixed role assigned: Administrator, Operator or Auditor. After the user identifies and authenticates to the module, the associated role is automatically assigned to the user. Concurrent users of the module are not supported (only one management console can be attached to the module through the serial port).

The Super Administrator role is assigned when two or more users with the Administrator role are authenticated to the module. The Super Administrator role is then assigned to the last user that successfully authenticated.

When the module is initialized the first time, as there are no users provided by default, the Super Administrator is assigned to the user that accesses the module to initialize it.

The User Application role is adopted implicitly by the external entities (user applications) after they identify and authenticate to the module and before requesting cryptographic services.

The following table describes the authorized roles and the services they can perform. Notice that there are some services that do not require an authorized role.

Role	Description
Super Administrator (SA)	This role is acquired by the user of the module when two or more users with the administrator role have authenticated into the module. The last user that authenticates to the module adopts this role. The Super Administrator role (SA) can perform authority management (add and delete management users), key management (create and import LMK, asymmetric and symmetric keys), and backup and restore services. Also, the Super Administrator can perform management services authorized to the Administrator and Operator roles.
Administrator (ADM)	This role is assigned when a user with this role authenticates successfully to the module. The module supports three users with this role. The administrator role (ADM) can perform key management (delete asymmetric and symmetric keys) and user management (add and delete users). Also, the Administrator can perform management services authorized to the Operator role.
Auditor (AUD)	This role is assigned when a user with this role authenticates successfully to the module. The module supports only one user with this role. The auditor role (AUD) can perform only audit management services (view management logs).
Operator (OP)	This role is assigned when a user with this role authenticates successfully to the module. The module supports only one user with this role. The operator role (OP) can perform system management (modify device maintenance information, network configuration and financial parameters), key management (import the SPK and view status of all keys), service management (view and modify the service configuration and the white list of authorized IP addresses, start, restart and stop services).

Role	Description
User Application (UA)	This role is assigned when an external entity (user application) authenticates successfully to the module via a network connection (TCP/IP). The User Application role can request cryptographic services to the module.

Table 4 – Roles

3.2. Services

The HSM provides two types of services: management services and cryptographic server services.

Management services include all services that can be executed from the management console or directly attached USB keyboard and screen by users of the module with the Administrator, Super Administrator, Operator and Auditor roles. These services are classified in the following groups:

- System Management
- Authority Management
- Key Management
- Backup and Recovery
- Service management
- User Application Management

Cryptographic server services are provided to applications that authenticates to the HSM through a network connection.

- Authentication services
- Symmetric key services
- Asymmetric key services
- Other services

3.2.1. Services in FIPS mode of operation

Table 5 below shows the management services that can be requested in FIPS mode of operation, including the cryptographic algorithms involved, the authorized roles that can execute them, and the required access to keys and CSPs. Services that do not require an authorized role are marked as “N/A” (not applicable).

Service	Algorithms	Roles				Keys/CSP [1]	Access
		SA	ADM	OP	AUD		
System Management							
View device information		N/A					
View device operating information		N/A					

[1] All services involving manipulation of keys or CSPs access the System Protection Key (SPK) with the read privilege.

Service	Algorithms	Roles				Keys/CSP [1]	Access
		SA	ADM	OP	AUD		
View device maintenance information		N/A					
Modify device maintenance information		✓	✓	✓			
View network configuration		N/A					
Modify network configuration		✓	✓	✓			
Apply network configuration		✓	✓	✓			
View financial parameters		N/A					
Modify financial parameters		✓	✓	✓			
View FIPS mode		N/A					
Set FIPS mode		✓	✓			All keys/CSPs	Zeroize
View management logs					✓		
Authority Management							
View login status		N/A					
User login	RSA Signature Verification	N/A					
User logout		N/A					
Modify USB token PIN		✓	✓	✓	✓		
Add administrator		✓				ADM's RSA public key	Input
Delete administrator		✓				ADM's RSA public key	Zeroize
Add operator		✓				OP's RSA public key	Input
Update operator		✓				OP's RSA public key	Zeroize
Add auditor		✓				AUD's RSA public key	Input
Update auditor		✓				AUD's RSA public key	Zeroize
Key Management							

Service	Algorithms	Roles				Keys/CSP [1]	Access
		SA	ADM	OP	AUD		
System Protection Key initialization	DRBG	N/A				All keys SPK SPK components Temp RSA public key	Zeroize Create Create Read
Import System Protection Key	RSA private key decryption	✓	✓	✓		Temp RSA key pair SPK components SPK	Create Read Create
Set random Local Master Key (LMK)	DRBG, KDF	✓				Root LMK LMKs	Create Create
Import Local Master Key (LMK)	RSA private key decryption, KDF	✓				LMK components Root LMK LMKs	Read Create Create
View LMK check value		N/A				Root LMK	Read
Generate RSA key pair	DRBG	✓				RSA key pair	Create
Delete RSA key pair		✓	✓			RSA key pair	Zeroize
View RSA key status		✓	✓	✓		RSA key pair	Read
Generate ECDSA key pair	DRBG	✓				ECDSA key pair	Create
Delete ECDSA key pair		✓	✓			ECDSA key pair	Zeroize
View ECDSA key status		✓	✓	✓		ECDSA key pair	Read
Generate DSA key pair	DRBG	✓				DSA key pair	Create
Delete DSA key pair		✓	✓			DSA key pair	Zeroize
View DSA key pair status		✓	✓	✓		DSA key pair	Read
Generate symmetric key	DRBG	✓				Symmetric key	Create
Import symmetric key	RSA private key decryption	✓				Symmetric key	Create
Delete symmetric key		✓	✓			Symmetric key	Zeroize

Service	Algorithms	Roles				Keys/CSP [1]	Access
		SA	ADM	OP	AUD		
View symmetric key status		✓	✓	✓		Symmetric key	Read
View symmetric key check value		✓	✓	✓		Symmetric key	Read
Backup and Recovery							
Backup cryptographic module	DRBG KTS	✓				Backup key Backup key components	Create Create
Restore cryptographic module	DRBG KTS	✓				Backup key components Backup key	Read Create
Service Management							
View service status		N/A					
View service configuration		✓	✓	✓			
Modify service configuration		✓	✓	✓			
White list management		✓	✓	✓			
Start service		✓	✓	✓			
Restart service		✓	✓	✓			
Stop service		✓	✓	✓			
User Application Management							
Add user application account		✓	✓			User application password	Write Import
Delete user application account		✓	✓			User application password	Delete
Other services							
On-demand self-tests (by resetting the module)	AES, DRBG, DSA, ECDSA, HMAC, RSA, SHS, Triple-DES	N/A					

Table 5 – Management services in FIPS mode of operation

Table 6 below shows the cryptographic server services that can be requested in FIPS mode of operation, the involved cryptographic algorithms and the access required to keys and CSPs. Each service indicates the service ID within parentheses. All services, with the exception of “User Application Authentication (ZA)”, require the User Application role in order to be executed. This role is obtained after the external entity authenticates successfully to the module (using the “User Application Authentication” service). The KTS entry in the algorithm table means that the service uses key wrapping to input or output CSPs in encrypted form. See sections 6.4 and 6.5 for more information.

Service	Algorithms	Access	Keys/CSP
Authentication Services			
User Application Authentication (ZA)	SHA-256	Read	User application password
Symmetric key services			
Key generation (A0)	AES	Create	AES key
	Triple-DES	Create	Triple-DES key
	KTS	Read	LMK
Data encryption (DF)	AES in ECB, CBC, CFB, CTR, OFB, XTS modes	Read	AES key
	Triple-DES in ECB, CBC modes	Read	Triple-DES key
	KTS	Read	LMK
Data decryption (DH)	AES in ECB, CBC, CFB, CTR, OFB, XTS modes	Read	AES key
	Triple-DES in ECB, CBC, CFB, OFB modes	Read	Triple-DES key
	KTS	Read	LMK
AES GCM Data encryption (TO)	AES in GCM mode	Read	AES key
	KTS	Read	LMK
AES GCM Data decryption (TP)	AES in GCM mode	Read	AES key
	KTS	Read	LMK
AES CCM Data encryption (TR)	AES in CCM mode	Read	AES key
	KTS	Read	LMK
AES CCM Data decryption (TS)	AES in CCM mode	Read	AES key
	KTS	Read	LMK

Service	Algorithms	Access	Keys/CSP
CMAC generation and verification (TQ)	AES	Create	AES key
	Triple-DES	Create	Triple-DES key
	KTS	Read	LMK
HMAC generation (XE)	HMAC with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	Read	HMAC key
	KTS	Read	LMK
Asymmetric key services			
RSA key pair generation (EI)	RSA, DRBG	Create	RSA private key
	KTS	Read	LMK
RSA public key export (EJ)	RSA	n/a	RSA public key
RSA public key encryption (ER)	RSA	n/a	RSA public key
RSA private key decryption (EP)	RSA	Read	RSA private key
	KTS	Read	LMK
RSA digital signature generation (EW)	RSA	Read	RSA private key
	KTS	Read	LMK
RSA digital signature verification (EY)	RSA	n/a	RSA public key
ECDSA key pair generation (TA)	ECDSA, DRBG	Create	ECDSA private key
	KTS	Read	LMK
ECDSA public key export (TH)	ECDSA	n/a	ECDSA public key
ECDSA digital signature generation (TB)	ECDSA	Read	ECDSA private key
	KTS	Read	LMK
ECDSA digital signature verification (TC)	ECDSA	n/a	ECDSA public key
ECDSA public key validation (TD)	ECDSA	n/a	ECDSA public key
DSA key pair generation (TE)	DSA, DRBG	Create	DSA private key
	KTS	Read	LMK
DSA digital signature generation (TF)	DSA	Read	DSA private key

Service	Algorithms	Access	Keys/CSP
	KTS	Read	LMK
DSA digital signature verification (TG)	DSA	n/a	DSA public key
Other services			
Message digest (3C, XG)	SHA-1, SHA-2, SHA-3	n/a	None
Random number generation (DR)	CTR_DRBG with AES256, Hash_DRBG with SHA-256	Read, Update	Entropy input string, Internal state
Get FIPS status (XA)		n/a	None

Table 6 – Cryptographic server services in FIPS mode of operation

3.2.2. Services in non-FIPS mode of operation

Table 7 below shows the management services that can be requested in non-FIPS mode of operation, including the cryptographic algorithms involved, the authorized roles that can execute them, and the required access to keys. Services that do not require an authorized role are marked as "N/A" (not applicable).

Service	Algorithms	Roles				Keys [2]	Access
		SA	ADM	OP	AUD		
Key Management							
Generate SM2 key pair		✓				SM2 key pair	Create
Delete SM2 key pair		✓	✓			SM2 key pair	Zeroize
View SM2 key status		✓	✓	✓		SM2 key pair	Read
Generate symmetric key	DRBG	✓				2-key Triple-DES	Create
Delete symmetric key		✓	✓			2-key Triple-DES	Zeroize

Table 7 – Management Services in non-FIPS mode of operation

Table 8 below shows the cryptographic server services that can be requested in non-FIPS mode of operation, the involved cryptographic algorithms and the access required to keys. Each service indicates the service ID within parentheses. All services require the User Application role in order to be executed. This role is obtained after the external entity authenticates successfully to the module (using the "User Application Authentication" service).

[2] All services involving manipulation of keys access the System Protection Key (SPK) with the read privilege.

Service	Algorithms	Access	Keys
Symmetric key services			
Key generation (A0)	DES, 2-key Triple-DES, SM4	Create	Symmetric key
Key generation (SE)	RC2, RC4, RC5, SEED, CAST, ARIA	Create	Symmetric key
Data encryption (DF)	2-key Triple-DES, SM4	Read	Symmetric key
Data decryption (DH)	SM4	Read	Symmetric key
Data encryption (SF)	RC2, RC4, RC5, SEED, CAST, ARIA	Read	Symmetric key
Data decryption (SH)	RC2, RC4, RC5, SEED, CAST, ARIA	Read	Symmetric key
CMAC generation and verification (TQ)	2-key Triple-DES	Read	Symmetric key
HMAC generation (XE)	HMAC key size less than 112 bits.	Read	Symmetric key
MAC generation (M0)	Triple-DES, AES, SM4, ISO9797 modes 1 and 3	Read	Symmetric key
Asymmetric key services			
RSA key pair generation (EI)	Key size less than 2048 bits.	Create	RSA private key
RSA public key encryption (ER)	Key size less than 2048 bits.	n/a	RSA public key
RSA private key decryption (EP)	Key size less than 2048 bits.	Read	RSA private key
RSA digital signature generation (EW)	RSA with SHA-1 Key size less than 2048 bits.	Read	RSA private key
RSA digital signature verification (EY)	Key size less than 1024 bits.	n/a	RSA public key
SM2 key pair generation (SI)	SM2, DRBG	Create	SM2 private key
SM2 public key export (SJ)	SM2	n/a	SM2 public key
SM2 public key encryption (SR)	SM2	n/a	SM2 public key
SM2 private key decryption (SP)	SM2	Read	SM2 private key
SM2 digital signature generation (SW)	SM2	Read	SM2 private key
SM2 digital signature verification (SY)	SM2	n/a	SM2 public key
ECIES public key encryption (SM)		n/a	ECIES public key
ECIES private key decryption (SN)		Read	ECIES private key

Service	Algorithms	Access	Keys
ECDSA key pair generation (TA)	P-192, K-163, B-163, Brainpool R1 and T1 curves	Create	ECDSA private key
ECDSA digital signature generation (TB)	P-192, K-163, B-163, Brainpool R1 and T1 curves	Read	ECDSA private key
ECDSA digital signature verification (TC)	Brainpool R1 and T1 curves	n/a	ECDSA public key
ECDSA public key validation (TD)	Brainpool R1 and T1 curves	n/a	ECDSA public key
DSA key pair generation (TE)	Key size less than 2048 bits.	Create	DSA private key
DSA digital signature generation (TF)	DSA with SHA-1 DSA with L=1024, N=160	Read	DSA private key
KCDSA key pair generation (SG)	KCDSA, DRBG	Create	KCDSA private key
KCDSA digital signature generation (SK)	KCDSA	Read	KCDSA private key
KCDSA digital signature verification (SL)	KCDSA	n/a	KCDSA public key
EC Diffie-Hellman shared secret computation (SB)		Create, Read	EC Diffie-Hellman public/private keys
Diffie-Hellman key generation (SC)		Create, Read	Diffie-Hellman private components
Diffie-Hellman shared secret computation (SD)		Create, Read	Diffie-Hellman private components
Other services			
Message digest (3C)	SM3	n/a	None

Table 8 – Cryptographic server services in non-FIPS mode of operation

3.3. Algorithms

The module implements cryptographic algorithms in two separate components:

- the CSM library (version v3.02.0025), a firmware component that implements general purpose cryptographic algorithms used for all cryptographic services.
- the Protection Card (version P52.0.00.0001), a hardware component that implements algorithms for internal usage (AES for key protection and RSA Signature Verification for user authentication).

The algorithms implemented in the module that are approved to be used in FIPS mode of operation are tested and validated by the CAVP.

The following tables show the cryptographic algorithms that are approved and allowed in FIPS mode of operation. Algorithms implemented in the protection card component include a reference.

Algorithm	Standard	Mode / Method	Key size	Use	CAVP Cert#
AES	[FIPS197] [SP800-38A]	ECB, CBC, CTR, OFB, CFB1, CFB8, CFB128	128, 192 and 256 bits	Data Encryption and Decryption	A955
		ECB	256 bits	Key Wrapping	
	[SP800-38B]	CMAC	128, 192 and 256 bits	MAC Generation and Verification	A960
	[SP800-38C]	CCM	128, 192 and 256 bits	Data Encryption and Decryption	A961
	[SP800-38D]	GCM	128, 192 and 256 bits	Data Encryption and Decryption	A965
	[SP800-38E]	XTS	128 and 256 bits	Data Encryption and Decryption	A959
AES (Protection Card)	[FIPS197] [SP800-38A]	ECB	256 bits	Encryption and Decryption of keys and CSPs	A1011
CKG[3]	[SP800-133]	DRBG	AES: 128, 192 and 256 bits Triple-DES: 192 bits	Symmetric Key Generation	Vendor affirmed
DSA	[FIPS 186-4]		L=2048, N=224; L=2048, N=256; L=3072, N=256	Key Pair Generation	A968
			L=2048, N=224; L=2048, N=256; L=3072, N=256	Domain Parameter Generation	
		SHA-224, SHA-256, SHA-384, SHA-512	L=2048, N=224; L=2048, N=256; L=3072, N=256	Digital Signature Generation	
			L=1024, N=160; L=2048, N=224; L=2048, N=256; L=3072, N=256	Domain Parameter Verification	
		SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	L=1024, N=160; L=2048, N=224; L=2048, N=256; L=3072, N=256	Digital Signature Verification	

[3] Asymmetric key generation is claimed in the corresponding entries for DSA, ECDSA and RSA algorithms.

Algorithm	Standard	Mode / Method	Key size	Use	CAVP Cert#
DRBG	[SP800-90A]	Hash_DRBG SHA-256 with Prediction Resistance (PR)	n/a	Random Number Generation	A963
		CTR_DRBG AES-256 without Derivation Function (DF), with PR			
ECDSA	[FIPS186-4]		P-224, P-256, P-384, P-521 K-233, K-283, K-409, K-571 B-233, B-283, B-409, B-571	Key Pair Generation	A967
		SHA-224, SHA-256, SHA-384, SHA-512	P-224, P-256, P-384, P-521 K-233, K-283, K-409, K-571 B-233, B-283, B-409, B-571	Signature Generation	A967
				Signature Generation without hashing	CVL. A967
			P-192, P-224, P-256, P-384, P-521 K-163, K-233, K-283, K-409, K-571 B-163, B-233, B-283, B-409, B-571	Public Key Verification	A967
		SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	P-192, P-224, P-256, P-384, P-521 K-163, K-233, K-283, K-409, K-571 B-163, B-233, B-283, B-409, B-571	Signature Verification	A967
				Signature Verification without hashing	CVL. A967
ENT (NP)	[SP800-90B]	n/a	n/a	Entropy Source for DRBG	n/a
HMAC	[FIPS198-1]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	112 bits or greater	Message Authentication Code	A962
KBKDF	[SP800-108]	HMAC SHA-256	256 bits	Key Derivation	A964
KTS	[SP800-38F]	AES (ECB) and HMAC SHA-256	256 bits	Key Wrapping	A955 A962
RSA	[FIPS186-4]	X9.31	2048, 3072 and 4096 bits	Key Pair Generation	A966

Algorithm	Standard	Mode / Method	Key size	Use	CAVP Cert#
		X9.31 with SHA-224, SHA-256, SHA-384, SHA-512	2048, 3072 and 4096 bits	Digital Signature Generation	
		X9.31 with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1024, 2048, 3072 and 4096 bits	Digital Signature Verification	
		PKCS#1v1.5 with SHA-224, SHA-256, SHA-384, SHA-512	2048, 3072 and 4096 bits	Digital Signature Generation	
		PKCS#1v1.5 with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1024, 2048, 3072 and 4096 bits	Digital Signature Verification	
		PSS with SHA-224, SHA-256, SHA-384, SHA-512	2048 and 3072 bits	Digital Signature Generation	
		PSS with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1024, 2048, and 3072 bits	Digital Signature Verification	
RSA (Hybrid)	[FIPS186-4]	PKCS#1v1.5 with SHA-256	2048 bits	Digital Signature Verification for User Authentication (hash value of the message is calculated by the firmware, and the Protection Card calculates the digital signature on the hashed value).	A1008

Algorithm	Standard	Mode / Method	Key size	Use	CAVP Cert#
SHS	[FIPS180-4]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	n/a	Message Digest	A954
SHA-3	[FIPS 202]	SHA3-224, SHA3-256, SHA3-384, SHA3-512	n/a	Message Digest	A953
Triple-DES	[SP800-67] [SP800-38A]	ECB, CBC, OFB, CFB1, CFB8, CFB128	192 bits (effective security strength is 112 bits)	Data Encryption and Decryption	A956
	[SP800-67] [SP800-38B]	CMAC	192 bits (effective security strength is 112 bits)	MAC Generation and Verification	A960

Table 9 – FIPS-Approved cryptographic algorithms

Algorithm / Keys	Caveat	Use
RSA PKCS#1 v1.5 Key Encapsulation[4] with 2048-bit keys	Provides 112 bits of encryption strength.	Key Establishment; allowed by IG D.9 in [FIPS140-2_IG].

Table 10 – FIPS-Allowed cryptographic algorithms

The table below shows the usage and key sizes not allowed in FIPS mode of operation.

Algorithm / Keys	Notes
Two-key Triple-DES	Not allowed per [SP800-131A].
SHA-1	Not allowed for Digital Signature Generation per [SP800-131A].
HMAC with key size less than 112 bits.	Not allowed for Message Authentication Code per [SP800-131A].
RSA keys of less than 2048 bits.	Not allowed for Key Generation, Digital Signature Generation, and Key Encapsulation per [SP800-131A].
RSA keys of less than 1024 bits.	Not allowed for Digital Signature Verification per [SP800-131A].

[4] RSA key encapsulation and RSA key wrapping are terms used interchangeably.

Algorithm / Keys	Notes
DSA keys of less than L=2048, N=224.	Not allowed for Key Pair Generation, Domain Parameters Generation, and Digital Signature Generation per [SP800-131A].
Elliptic curves P-192, K-163 and B-163.	Not allowed for Key Pair Generation, Domain Parameters Generation, and Digital Signature Generation per [SP800-131A].
Brainpool R1 and T1 elliptic curves.	Non-approved elliptic curves.

Table 11 – Algorithm usage and key sizes not allowed in FIPS mode of operation

The table below shows the cryptographic algorithms implemented in the module that are not allowed in FIPS mode of operation.

Algorithms	Description
KCDSA	Korean Certificate-based Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
SM2	Chinese Elliptic Curve Digital Signature Algorithm
SM3	Chinese Message Digest algorithm
SM4	Chinese Block Cipher Symmetric algorithm
RC2, RC4, RC5, SEED, CAST, ARIA, DES	Symmetric encryption/decryption algorithms
Triple-DES MAC and ISO9797 modes 1 and 3	Message Authentication Code algorithms
Diffie-Hellman, EC Diffie-Hellman	Shared secret computation algorithms, key generation

Table 12 – Non-Approved cryptographic algorithms

3.4. Identification and Authentication

The module uses identity-based authentication to identify and authenticate users and user applications in the module:

- For Super Administrator, Administrator, Operator or Auditor roles, authentication is performed using a challenge-response mechanism with the user's credentials (a 2048-bit RSA key pair) stored in the user's USB token.
- For external entities (user applications), these assume the User Application role, and the authentication is performed using a challenge-response mechanism with the user's ID and password, which are stored in the module (the password is stored in encrypted form).

3.4.1. User authentication

The module identifies and authenticates users of the module through the following steps:

1. The user inserts the USB token into the USB port, and enters the PIN through the module's console.
2. The module authenticates against the USB token with the PIN provided by the user.
3. The module generates a challenge consisting of a 256-bit random number using the SP800-90A DRBG and sends it to the USB token.
4. The USB token generates a digital signature of the challenge (using the user's RSA private key stored in the USB token) and sends it back to the module.

5. The module performs signature verification of the challenge with the user's public key (already stored in the module).

If authentication succeeds, the user adopts the role assigned during its creation.

When a new user of the module is added, the module first assigns an ID and the desired role (Administrator, Operator, or Auditor). The module then asks for the insertion of the new user's USB token and a PIN of eight digits (to allow access to the token) and requests the USB token to generate credentials (a 2048-bit RSA key pair) for the new user. The module then imports and stores the user's RSA public key together with the ID and the associated role of the user.

The module is provided from the factory uninitialized, that is, with no users or data. The first time a user connects to the module, authentication is not enforced and the user implicitly adopts the Super Administrator role. The user performs the initialization of the module and adds the first user with the Administrator's role.

The PIN provided by the user has the purpose of verifying the identity of the owner of the USB token, and not for authenticating the module. The USB token is physically possessed by the user and outside the boundary of the module.

A user remains authenticated until the user logs out from the module or the module is powered off (no authentication data remains in the module).

3.4.2. External entity authentication

External entities (e.g., server applications) need to identify and authenticate to the module when establishing the network connection and before requesting cryptographic services.

The authentication mechanism works as follows:

1. The external entity sends the authentication request, including the user id and a challenge of 128 bits generated randomly.
2. The module receives the authentication request and sends back a response, including a 128-bit challenge generated randomly by using the SP800-90A DRBG.
3. The external entity calculates a SHA-256 message digest using as input the concatenation of both challenges and the password in plaintext format. The external entity sends the second authentication request message.
4. The module decrypts the password (stored by the module in encrypted form) corresponding to the user id initially sent by the request; calculates the same SHA-256 message digest using as input the concatenation of both challenges and the password in plaintext, and verifies the received hash matches the one calculated locally.
5. If both values match, then authentication succeeds, and the external entity adopts the User Application role. Otherwise, authentication fails.

The password is eight characters long and can contain any character that can be input through the keyboard.

When a new external entity is added to the module, the module stores the user ID and password provided by the user.

An external entity remains authenticated only during the life span of the network session, or until the module is powered off (no authentication data remains in the module).

3.5. Authentication strength

The user authentication mechanism uses a 2048-bit RSA key pair. According to [SP800-57], such a key provides a security strength of 112 bits. Therefore, the probability of a successful authentication by guessing the private key, using a fake USB token with a non-registered user's credential, is $2^{-112} \approx 10^{-33}$, which is far less than the maximum probability of 10^{-6} required by the FIPS 140-2 standard. In addition, considering that the authentication process requires entering the PIN manually through the module's console and assuming the user could perform a maximum of 1 attempt per second, the total probability of guessing the credentials in one minute is $10^{-33} * 60 \approx 10^{-31}$. This number is still far less than the maximum probability of 10^{-5} required by the FIPS 140-2 standard.

The external entity authentication mechanism uses an eight-character password. Considering a minimum alphabet of 36 symbols (numbers and alphabetic characters), the password still yields $36^8 \approx 10^{12}$ possible combinations. In this case, the probability of success of random attempts is close to 10^{-12}. This number is less than the maximum probability of 10^{-6} required by the FIPS 140-2 standard. Considering that authentication is performed through a network connection, with an estimation of a throughput of 10000 authentication messages per second, the probability of success of random attempts in a minute is $36^{-8} \times 10000 \times 60 \approx 10^{-6}$, which still is less than the maximum probability of 10^{-5} required by the FIPS 140-2 standard.

4. Physical Security

This section describes the physical security mechanisms that the module employs in order to restrict unauthorized physical access to the contents of the module and to deter unauthorized use or modification of the module.

4.1. Static Protection

All components of the HSM are enclosed in a 1.2-millimeter galvanized steel case, opaque to the visible spectrum. The case contains gaps at the cooling holes that are covered internally by a stainless steel perforated sheets and dustproof sponge acting as a wired net filter, which prevents visibility on the internals and probe into the module through these holes.

Physical ports in both the front and rear panels are fixed to the chassis from the inside of the case, so they cannot be detached.

The case has a removable cover on the top, which is fixed to the case through six screws. The junctures between the case and the cover are protected with two tamper-evidence seals located at both sides of the chassis. Another two seals cover the screws that fix the chassis handlers to the case for additional protection. The pictures below show the position of the seals in the HSM.

Figure 6 – Seal locations (left view)

Figure 7 – Seal locations (right view)

The HSM is delivered by Sansec with the four tamper-evidence seals applied at the positions shown in the figures above. Any attempt to remove the seals or open the case cover will leave evidence. Users of the HSM are responsible for regularly inspecting the seals and verifying that they remain intact and in the location shown in the User's Manual.

If evidence of tampering is detected, the module shall be considered non-compliant. A user with the Administrator role must initialize and return the HSM to Sansec in order to restore the tamper-evidence seals.

4.2. Dynamic Protection

During the operation of the HSM, keys and CSPs are stored in plaintext into the volatile memory. In order to prevent the disclosure of such sensitive information, the HSM includes a tamper detection switch. When the chassis cover is opened, the tamper switch is triggered and the HSM reboots, zeroizing all the information stored in the RAM including keys and

CSPs. After the HSM restarts, the Administrator always needs to import the System Protection Key (SPK) for decrypting and storing the keys and CSPs into the volatile memory in order to turn the HSM operational.

5. Operational Environment

5.1. Applicability

The module operates in a non-modifiable operational environment. Once the firmware of the module is loaded, it cannot be modified or erased. Therefore, FIPS 140-2 requirements for the operational environment are not applicable to the module.

6. Cryptographic Key Management

The following table summarizes the keys and CSPs that are used by the cryptographic services implemented in the module:

Name	Purpose	Generation	Entry and Output	Storage	Zeroization
System Protection Key (SPK) 256-bit AES key	Protection of keys and CSPs stored in the module.	During HSM initialization, using the SP800-90A DRBG.	Input from USB tokens during SPK loading (key encapsulation). Output to USB tokens during key initialization (key encapsulation).	n/a (SPK in RAM after key entry; SPK key components are stored in USB tokens).	n/a (module does not store the SPK).
Root Local Master Key (LMK) 256-bit AES key	Key derivation of LMKs.	During HSM initialization using the SP800-90A DRBG.	Input from USB tokens (key encapsulation).	Encrypted with SPK.	Zeroized when HSM is initialized.
Local Master Keys (LMK) 256-bit AES keys	Key wrapping of user application's keys	During HSM initialization. Derived from the root LMK using SP800-108 KDF with HMAC SHA256	n/a	Encrypted with SPK.	Zeroized when HSM is initialized.
Temporary 2048-bit RSA key pair	Key encapsulation of SPK and root LMK components.	Whenever a new key transport is started; generated compliant with FIPS 186-4 and using the SP800-90A DRBG.	Only RSA public key is output to the USB token.	n/a (only in RAM)	Zeroized from RAM after key transport ends.
Crypto Officer RSA public key	Identity-based Authentication of the HSM.	None (RSA key pair is generated by the USB token).	RSA public key is input from the USB token.	In plaintext form.	Zeroized when user is deleted or HSM is initialized.
Crypto Officer PIN	Authentication of USB token.	None	Input by Crypto Officer from console. Output to USB token.	n/a (only in RAM, stored in USB token).	Zeroized once user authenticates
User Triple-DES keys User AES keys (128, 192 and 256 bits)	Protection of user data.	By a service request (server application) or a key management service (Crypto Officer). Generated using the SP800-90A DRBG.	Input and output as part of service requests (key wrapping). Input as part of key management services from USB tokens (key encapsulation). Output through management console (key wrapping).	Encrypted with SPK.	Zeroized when Crypto Officer deletes the key through key management services.

User HMAC keys	Message Authentication	None	Input as part of service requests (key wrapping).	n/a (only in RAM).	Zeroized when the service request is finished.
User ECDSA key pair	Digital Signature generation and verification.	By a service request (server application) or a key management service (Crypto Officer). Generated compliant with FIPS 186-4 and using the SP800-90A DRBG.	Input and output as part of service requests (key wrapping).	Encrypted with SPK.	Zeroized when Crypto Officer deletes the key pair through key management services.
User RSA key pair					
User DSA key pair					
Backup Key	Protection of CSPs exported form and imported to HSM	During the backup key management service. Generated using the SP800-90A DRBG	Input from user's USB tokens during key restore (split knowledge). Output to user's USB tokens during key backup (split knowledge).	n/a (only in RAM, key components are stored in user's USB tokens.)	Zeroized when backup or restore operation is complete.
User Application password	Identity-based authentication of user applications.	n/a	Input by user with Administrator or Super Administrator role from console.	Encrypted with SPK.	Zeroized when user is deleted or HSM is initialized.
Entropy input string	Compose DRBG internal state.	Obtained from NRBG	n/a	n/a (only in RAM).	n/a
DRBG internal state (V, C, Key)	DRBG internal state.	During DRBG initialization.	n/a	n/a (only in RAM).	Zeroized when DRBG is no longer used.

Table 13.1 – Life cycle of keys and critical security parameters (CSPs)

6.1. Random Number Generation

The module employs a Deterministic Random Bit Generator (DRBG) compliant with [SP800-90A] for the creation of symmetric and asymmetric keys, the creation of the AES GCM Initialization Vector, the creation of random number challenges for the identity-based authentication mechanism, and for processing the Random Number Generation service request. The DRBG supports the Hash_DRBG and CTR_DRBG mechanisms.

The DRBG is initialized during module initialization; the module loads by default the DRBG using the CTR_DRBG mechanism with AES-256, without derivation function and with prediction resistance. This mechanism is used for key generation, providing 256 bits of security strength.

The Hash_DRBG using SHA-256 and prediction resistance is provided as an alternative mechanism for the Random Number Generation service.

The module performs DRBG health tests as defined in Section 11.3 of [SP800-90A].

For seeding the DRBG, the module uses a Non-deterministic Random Bit Generator (NRBG) compliant with [SP800-90B]. The NRBG is implemented by the cryptographic module and therefore it is within its logical boundary. The NRBG provides full entropy to seed and reseed the DRBG, therefore the DRBG provides 256 bits of security strength.

The NRBG implements health tests as defined in Section 4.3 and Section 4.4 of [SP800-90B] on the output to ensure that consecutive random numbers do not repeat.

The NRBG utilizes a non-physical noise source, therefore the entropy source is indicated as ENT (NP) in table 9 and the certificate.

6.2. Key Generation

The module performs symmetric and asymmetric key generation for cryptographic service requests, key management services, and for key and CSP protection.

Triple-DES and AES symmetric keys are generated using random data from the SP800-90A DRBG. RSA, DSA and ECDSA key pairs are generated in compliance with [FIPS186-4], and also using the SP800-90A DRBG.

Note: in accordance with [FIPS140-2_IG] D.12, the module performs Cryptographic Key Generation (CKG) for symmetric and asymmetric keys as per [SP800-133] (vendor affirmed).

6.3. Key Derivation

The module performs key derivation of Local Master Keys (LMKs) from the root LMK during the key initialization invoked by the administrator. The module implements a key derivation function (KDF) using the HMAC-SHA-256 algorithm, in compliance with [SP800-108].

6.4. Key Transport

The module protects keys whenever they are input to or output from the module. The module implements the following key transport mechanisms:

- Key wrapping is used for protecting keys that are part of cryptographic service request or response messages. The module uses AES in ECB mode and HMAC-SHA-256 algorithms, compliant with [SP800-38F]. The keys used for this mechanism are the LMKs stored in the module, whose size is 256 bits.
- RSA key encapsulation is used for protecting key components that are transported to and from USB tokens. The module uses RSA public encryption and private decryption primitives, compliant with PKCS#1v1.5. The keys used for this mechanism have a key size of 2048-bit keys.

According to “Table 2: Comparable strengths” in [SP800-57], the key sizes of AES and RSA provide the following security strength:

- AES HMAC-SHA-256 key wrapping provides 256 bits of encryption strength.
- RSA key encapsulation provides 112 bits of encryption strength.

Note: the module also provides Diffie-Hellman and EC Diffie-Hellman services for key agreement, but they cannot be used in FIPS mode of operation.

6.5. Key Entry / Output

The module supports electronic distribution of keys in encrypted form. The module does not support intermediate key generation key output, and does not enter or output keys in plaintext format outside its physical boundary. The module also supports manual distribution of keys in encrypted form, with the exception of the backup key, which is input and output in plaintext using split knowledge procedures.

Cryptographic services requested by external entities may involve input of keys in the request message (e.g. data encryption or decryption, signature generation, HMAC) or output of keys in the response message (e.g. key generation). The module uses key wrapping with AES and HMAC-SHA256 as the key transport mechanism, using one of the LMKs stored in the module.

Keys can be also input from or output to external USB tokens through the key management services. In all the following cases, the keys are input or output in encrypted form, using RSA key encapsulation with a 2048-bit key as the key transport mechanism:

- The System Protection Key (SPK) is created during the initialization of the module and needs to be input into the module each time the module is powered up. The module uses split knowledge procedure to create three SPK components, exporting each of them to different, ad-hoc USB tokens. Every time the module is power-up, at least two SPK components have to be input for the module to operate.
- The root Local Master Key (LMK) can be input during the initialization of the module. The key is input using three separate key components stored in different, ad-hoc USB tokens.
- User's symmetric keys (AES and Triple-DES) can also be imported during the operation of the module. The keys are input using several (from two to nine) key components stored in ad-hoc USB tokens.

The backup key is created to encrypt the backup archive before transferring all the keys out of the module. The module uses a split knowledge procedure to create three key backup components, exporting each of them in plaintext form to the USB token pertaining to each of the users with the Administrator role, who have to authenticate to the module first. In order to restore the keys in the module, two backup key components have to be input to compose the backup key. Users with the Administrator role have to authenticate before input each key component.

User's symmetric keys (AES and Triple-DES) that are imported to the module can be also output through the console. Keys are shown encrypted using the AES in ECB mode and HMAC-SHA-256 key wrapping. The HMAC value is also shown.

The RSA public key pertaining to the user of the module is input in plaintext form when the user is added to the module from the user's USB token.

6.6. Split Knowledge Procedure

The module uses the split knowledge procedure for entry and output of the System Protection Key (SPK) and the Back-up Key. This mechanism is based on Shamir's Secret Sharing algorithm. The module splits a key into three components, which are stored separately in three different USB tokens. Any two of these three components can be entered to the module in order to reconstruct the original key.

6.7. Key / CSP Storage

All keys and CSPs are stored in encrypted form into the non-volatile memory or the file system.
The module protects keys and CSPs using the same mechanism used for key wrapping (AES in ECB mode and HMAC-SHA-256) compliant with [SP800-38F]. The calculated HMAC value is stored with the encrypted key and the integrity of keys and CSPs are verified during decryption.

All keys and CSPs are encrypted using the System Protection Key (SPK), which is not stored in the module and only remains in RAM while the module is operational.

6.8. Key / CSP Zeroization

All private and secret keys and CSPs are stored in encrypted form.

The "SPK initialization" service allows a user with the Administrator role to completely erase the contents of the module. This also happens when the module is switched from non-FIPS mode to FIPS mode, or viceversa, through the "Set FIPS mode" service. In both cases, all keys and CSPs are zeroized. In addition, a management service that implies the deletion of keys (key initialization, changing the mode of operation of the module or deleting a specific key), the module zeroizes the affected keys.

The zeroization function overwrites the storage of keys and CSPs with “zeros”.

7. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, FCC PART 15, Subpart B, Unintentional Radiators, Digital Devices, Class B (i.e. for home use).

8. Self Tests

8.1. Power-Up Tests

The module implements a series of power-up self-tests to ensure that cryptographic algorithms work as expected and the module has not been corrupted. Power-up self-tests include integrity tests on the firmware and cryptographic algorithm self-tests.

When a user powers on the module, power-up self-tests are executed automatically. While the module is executing self-tests, input and output are inhibited. Services are not available until all self-tests are completed successfully.

When the module finishes the power-up self-tests successfully, the LCD screen menu becomes responsive and the management console is enabled. If any of the power-up self-test fails, the module enters into the error state, and an error message is shown in the LCD screen. Input and output are inhibited and none of the management or cryptographic services is available. The user must restart the module.

8.1.1. Integrity Tests

The integrity of the module is verified by comparing a CRC32 value calculated at run time with the value stored in the module which was computed during the HSM production process. Firmware integrity covers all the programs and link libraries of the core components of the operating system, and the internal firmware program of the protection card.

If the CRC32 values do not match, the integrity test fails and the module enters the error state.

8.1.2. Cryptographic algorithm tests

The module performs self-tests on all FIPS-Approved cryptographic algorithms supported in the approved mode of operation, using the Known Answer Tests (KAT) and Pair-wise Consistency Tests (PCT) shown in the following table.

Algorithm	Test
AES	• KAT AES(ECB) with 128-bit key, encryption • KAT AES(ECB) with 128-bit key, decryption • KAT AES(CTR) with 128-bit, encryption • KAT AES(CTR) with 128-bit, decryption • KAT AES(CCM) with 128-bit key, encryption • KAT AES(CCM) with 128-bit key, decryption • KAT AES(GCM) with 128-bit key, encryption • KAT AES(GCM) with 128-bit key, decryption • KAT AES(CMAC) with 128-bit key • KAT AES(ECB) with 256-bit key, encryption (Protection Card) • KAT AES(ECB) with 256-bit key, decryption (Protection Card)
Triple-DES	• KAT Triple-DES(ECB) with 192-bit key, encryption • KAT Triple-DES(ECB) with 192-bit key, decryption • KAT Triple-DES(CMAC) with 192-bit key

Algorithm	Test
SHS	• KAT SHA-1 • KAT SHA-224 • KAT SHA-256 • KAT SHA-384 • KAT SHA-512 • KAT SHA3-224 • KAT SHA3-256 • KAT SHA3-384 • KAT SHA3-512
HMAC	• KAT HMAC-SHA-256
DSA	• PCT DSA with L=2048, N=256 and SHA-256, signature generation • PCT DSA with L=2048, N=256 and SHA-256, signature verification
ECDSA	• PCT ECDSA with P-256 and SHA-256, signature generation • PCT ECDSA with P-256 and SHA-256, signature verification • PCT ECDSA with K-233 and SHA-512, signature generation • PCT ECDSA with K-233 and SHA-512, signature verification • PCT ECDSA with B-571 and SHA-384, signature generation • PCT ECDSA with B-571 and SHA-384, signature verification
RSA	• KAT RSA PKCS#1v1.5 with 2048-bit key and SHA-256, signature generation • KAT RSA PKCS#1v1.5 with 2048-bit key and SHA-256, signature verification • KAT RSA with 2048-bit key, public-key encryption • KAT RSA with 2048-bit key, private-key decryption • KAT RSA PKCS#1v1.5 with 2048-bit key and SHA-256, signature verification (hybrid implementation using Protection Card)
DRBG	• KAT Hash_DRBG using SHA-256, with PR • KAT CTR_DRBG using AES-256, without DF
NRBG	• Repetition Count Test (RCT) and Adaptive Proportion Test (APT)
SP800-108 KBKDF	• KAT Counter mode with HMAC-SHA-256

Table 14 – Self-tests.

For KATs, the module calculates the result and compares it with the known value. If the answer does not match the known answer, the KAT fails and the module enters the Error state. For PCTs, if the signature generation or verification fails, the module enters the Error state.

8.2. On-Demand self-tests

On-Demand self-tests can be invoked by restarting the module, thus forcing the module to run the power-up self-tests.

8.3. Conditional Tests

The module performs conditional tests on the cryptographic algorithms using Pair-wise Consistency Tests (PCT) and Continuous Random Number Generator Test (CRNGT), as shown in the following table.

Algorithm	Test
DSA key generation	PCT using SHA-256, signature generation and verification.
ECDSA key generation	PCT using SHA-256, signature generation and verification.
RSA key generation	PCT using SHA-256, signature generation and verification. PCT using public key encryption and private key decryption.
NRBG	Repetition Count Test (RCT) and Adaptive Proportion Test (APT)

Table 15 – Conditional tests

Note: CRNGT on the SP800-90A DRBG is not required per IG 9.8 in [FIPS140-2_IG].

If a conditional test fails, the module enters into the error state, and an error message is shown in the LCD screen. Input and output are inhibited and none of the management or cryptographic services is available. The user must restart the module.

9. Guidance

In FIPS Approved mode of operation, the module must be operated using the FIPS approved services, with their corresponding FIPS approved or FIPS allowed cryptographic algorithms provided in this Security Policy (see Section 3.2). In addition, cryptographic algorithms and their key sizes must also comply with [SP800-131A].

In FIPS mode of operation, all rules above are enforced by the module. In case a service request does not meet any of the rules, the module rejects the request.

9.1. HSM initialization

The HSM is shipped to the vendor without any initialization of keys or CSPs. By default, the HSM is configured to operate in FIPS mode.

The first user of the module implicitly acquires the Super Administrator role, and is allowed to perform the HSM initialization without any authentication.

The user must perform the initialization of the HSM following the instructions included in chapter 4 "Installation and Device Management" of the Sansec HSM User's Manual [HSM-UM].

1. Verify that the external condition of the package to see if there are signs of damage, or if the package has been opened during transit.
2. Open the package and verify with the content list that the HSM and all accessories are included.
3. Verify that the four tamper evidence seals are intact and located at the expected positions (see Figure 6 and Figure 7).
4. Connect the HSM to a power supply, you can distinguish the AC/DC power from the power input inlets of the module.
5. Connect a PC to the HSM through the serial port, or connect VGA screen and USB keyboard to the VGA port and any of the USB port.
6. Power-up the HSM.
7. Login into the system and run the HSM management program (hsmm).
8. Verify that the product model and version provided in the "View Device Basic Information" menu option matches the following information:
    - Vendor: SANSEC
    - Product Mode: SecHSM V2-1
    - Product No.: (unique number)   Device Version: v3.02.0025.
9. Use the Installation Wizard to perform the following activities:
    - Initialize the device, create the System Protection Key (SPK) and export the SPK key components to USB tokens.
    - Create the HSM users for all roles (Administrators, Operator, Auditor) and generate their credentials in the USB tokens.
    - Generate (or import) the Local Master Key (LMK).
    - Generate (or import) symmetric keys (optional).
    - Generate RSA and ECDSA keys (optional).
    - Configure the network.
    - Configure the device information

- Configure the service startup parameters.
- Configure the IP address whitelist and users for the services.

9.2. USB Tokens

In order to initialize the HSM, the following Sansec USB tokens must be available:

- Five USB tokens for the generation of the credentials of each of the users (three administrators, one operator and one auditor).
- Three USB tokens for exporting the SPK key components.

The HSM uses the same model of USB tokens for storing keys and user's credentials. However, they are configured differently depending on their usage:

- For the creation and storage of a user's credential (a 2048-bit RSA key pair for each user with the Administrator, Operator and Auditor roles). These tokens also store the backup key components.
- For the storage of a key component for the LMK, SPK or a symmetric key.

The USB tokens must be initialized with an ad-hoc utility. Access to the USB token is protected through an eight-digit PIN.

9.3. Verification of Tamper Evidence Seals

On a periodic basis, users of the HSM must verify that the tamper evidence seals are intact and located in the expected positions of the chassis. If evidence of tampering is detected, the module shall be considered non-compliant. A user of the module with the Administrator role shall be informed, who shall conduct as described in Section 4.1.

9.4. Algorithm Considerations

9.4.1. AES GCM IV

For AES GCM encryption, the module generates a random, 96-bit initialization vector (IV), using the SP800-90A DRBG implemented in the module using the RBG-based construction method as defined in section 8.2.2 of [SP800-38D]. The module is compliant with scenario 2 of FIPS 140-2 IG A.5 in [FIPS140-2_IG].

9.4.2. AES XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in [SP800-38E]. In addition, the length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks, that is, 16 MiB of data. The module verifies that each key stored in the module is not used for data encryption beyond the limit of 2^{15} blocks by using a counter of the number of encryptions performed with each key since its generation.

For those keys provided by external entities as part of the cryptographic service requests, the verification of this limit must be enforced by the entities that request the service (e.g. server applications).

In addition, to meet the requirement in [FIPS140-2_IG] A.9, the module implements a check to ensure that the two AES keys used in XTS-AES algorithm are not identical.

9.4.3. Triple-DES Keys

Data encryption using the same three-key Triple-DES key shall not exceed 2^{16} Triple-DES blocks, in accordance to [SP800-67] and IG A.13 in [FIPS140-2-IG]. The module verifies that each key stored in the module is not used for data encryption beyond the limit of 2^{16} blocks by using a counter of the number of encryptions performed with each key since its generation.

For those keys provided by external entities as part of the cryptographic service requests, the verification of this limit must be enforced by the entities that request the service (e.g., server applications).

9.4.4. Key Establishment Methods

The key establishment methodology for the transport of keys between the module and the USB tokens (RSA key encapsulation with 2048-bit keys) provides 112 bits of encryption strength.

The key establishment methodology for the transport of keys between the module and the application (key wrapping with AES in ECB mode and HMAC-SHA-256) provides 256 bits of encryption strength.

10. Mitigation of Other Attacks

There are no mitigations from other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation System
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECIES	Elliptic Curve Integrated Encryption Scheme
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
KCDSA	Korean Certificate-based Digital Signature Algorithm
MiB	Mebibyte (a multiple of the unit byte for digital information)
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NRBG	Non-Deterministic Random Bit Generator
PCT	Pair-wise Consistency Test
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-2 **FIPS PUB 140-2 - Security Requirements For Cryptographic Modules**
May 2001
https://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

FIPS140-2_IG **Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program**
January 5, 2021
https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/fips1402ig.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
March 2012
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

HSM-UM **Sansec HSM User manual v2.1**
January 2021

HSM-CS **Sansec HSM Command Set Manual v1.2**
September 2018

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** February 2003
https://www.ietf.org/rfc/rfc3447.txt

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38B **NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**
May 2005
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38B.pdf

SP800-38C **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf

SP800-38D **NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**
November 2007
https://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-57	**NIST Special Publication 800-57 Part 1 Revision 5 - Recommendation for Key Management Part 1: General** May 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf
SP800-67	**NIST Special Publication 800-67 Revision 2 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** November 2017 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf
SP800-90A	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-90B	**NIST Special Publication 800-90A - Revision 1 - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP800-108	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions (Revised)** October 2009 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf
SP800-131A	**NIST Special Publication 800-131A Revision 2- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf
SP800-133	**NIST Special Publication 800-133 Revision 1- Recommendation for Cryptographic Key Generation** June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP800-135	**NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf