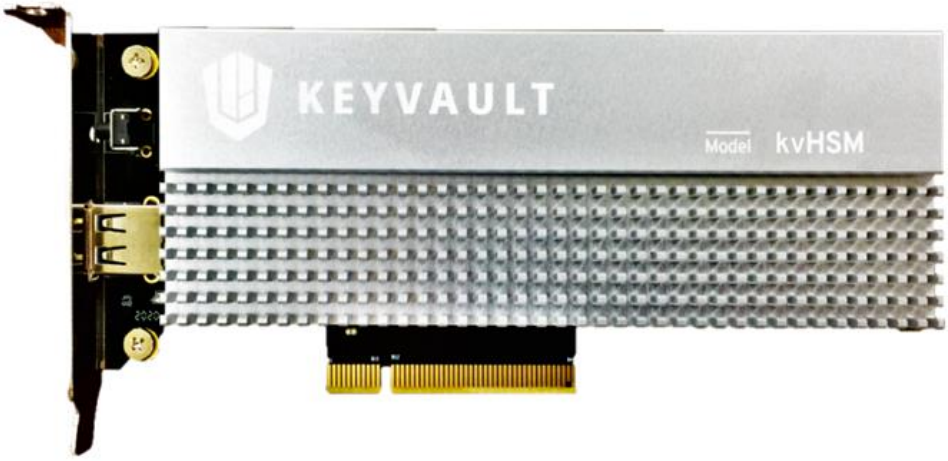
KeyVault Hardware Security Module (kvHSM)

Security Policy



Document: KeyVault Hardware Security Module Security Policy

Version: 2.0

Date: 2023/01/06

Phone: +886-2-29343166

WWW: www.wisecure-tech.com

Table of Contents

1. Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for the cryptographic module "KeyVault Hardware Security Module (kvHSM)" (hereafter referred to as "kvHSM"). The module 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 module is a hardware security module made by WiSECURE Technologies. The module is a PCIe card that stores and manages cryptographic keys as a vault. It manages the entire lifecycle of keys, including generation, distribution, storage, destruction and archiving, and provides data encryption, data decryption, signature generation, signature verification, message digest, message authentication code (MAC), random number generation and key management services.

The module provides a robust environment for cryptographic operations. It is designed in reaction to compromising, physical intrusion, and tampering. The module is carefully designed to mitigate risks of key leakage and dismisses the threat posed by side channel attack (SCA).

The host application connects to the module through PCIe interface. The host identifies and authenticates against the module. Once authentication succeeds, the host requests cryptographic services to the module, which processes the requests and sends back the result. In addition, administrators of the module can access the management functions after authenticated with admin USB tokens. The overall architecture is shown in Figure 1.

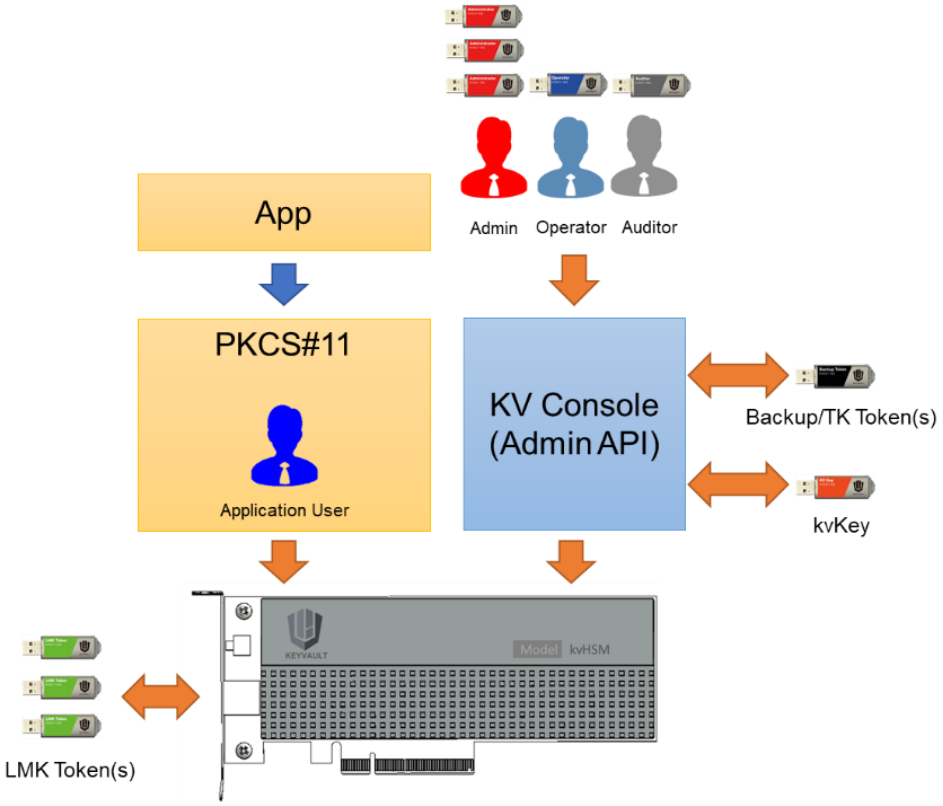


Figure 1 – Overall architecture

The module are defined as multi-chip embedded cryptographic modules as defined by FIPS 140-2. It meets overall FIPS 140-2 Level 3 requirement in FIPS mode and support major algorithms needed by the modern crypto-based applications.

The physical dimensions of the module are 167 mm x 68.9 mm x 16.5 mm (width x height x depth), as shown in Figure 2.

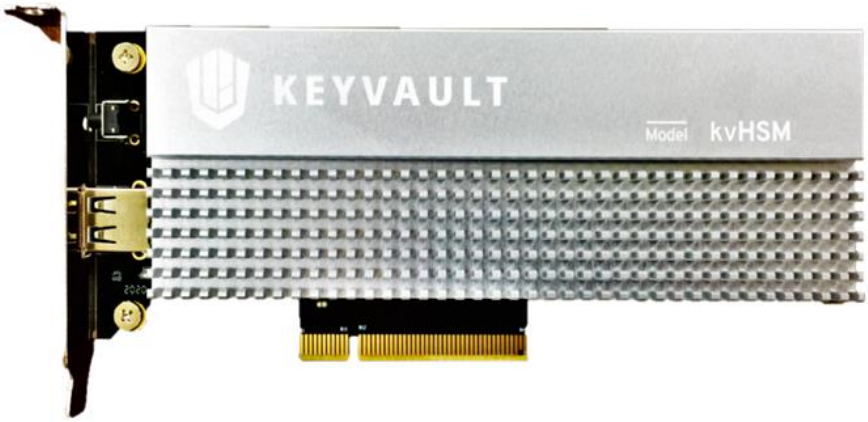


Figure 2 – Front view

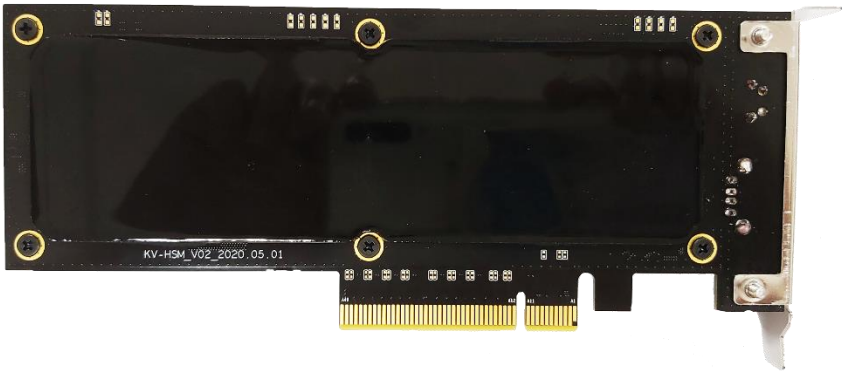


Figure 3 – Back view

The module provides a hardened, tamper-resistant environment. The module is covered with an aluminum enclosure. It covers all the major components and tamper response circuits that can detect the open event of the enclosure.

1.2. Cryptographic Module Description

For the purpose of the FIPS 140-2 validation, the module is a multi-chip embedded 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
9	Self-Tests	3
10	Design Assurance	3
11	Mitigation of Other Attacks	3
Overall Level		3

Table 1 – Security levels

The module is covered by aluminum enclosure, and the bottom side covered by epoxy material as shown in Figure 3. The physical boundary of the module is defined as the whole PCIe card, and the security boundary is red marked as shown in Figure 4.

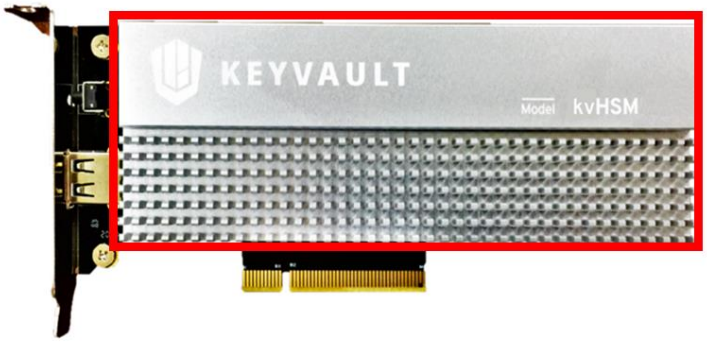


Figure 4 – Security boundary

The block diagram of the module is shown as Figure 5. It contains the FPGA, Controller, Secure Element, DDR3, SPI Flash, Battery-Backup SRAM, Photodiode, Detect Switch and Battery. The crypto engines are implemented in FPGA and in Secure Element. The Keys and CSP are stored in Secure Element and encrypted by the Local Master Key (LMK). The LMK is kept in the Battery-backup SRAM which will be zeroized by the enclosure open event.

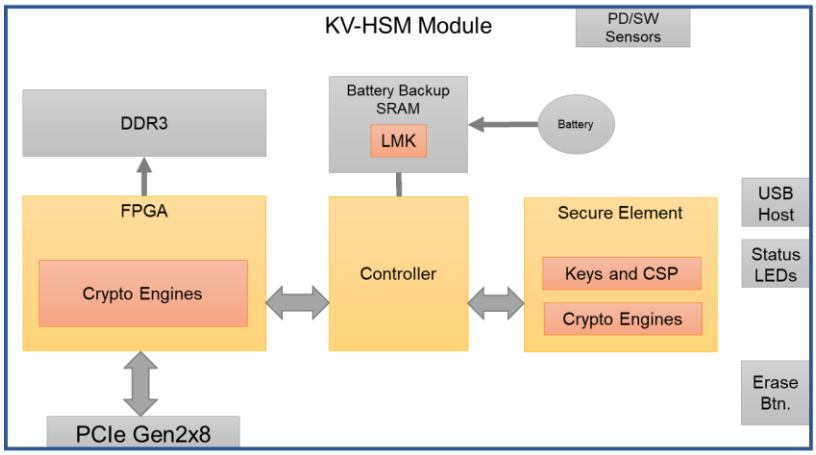


Figure 5 – Block diagram

1.3. Mode of Operation

The module supports two modes of operation.

- In "FIPS mode", (the Approved mode of operation) only approved or allowed security functions with enough security strength can be used. The FIPS mode of operation is implicitly assumed when a administrator initializes the module for the first time.
- 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.

Once the module is operational, Administrator can check the mode in kvConsole/Setup. The kvConsole is the external utility for operator to communicate and manage the module. It is out of module boundary and is not part of the validation. The kvConsole UI will display the mode of

operation of the module.

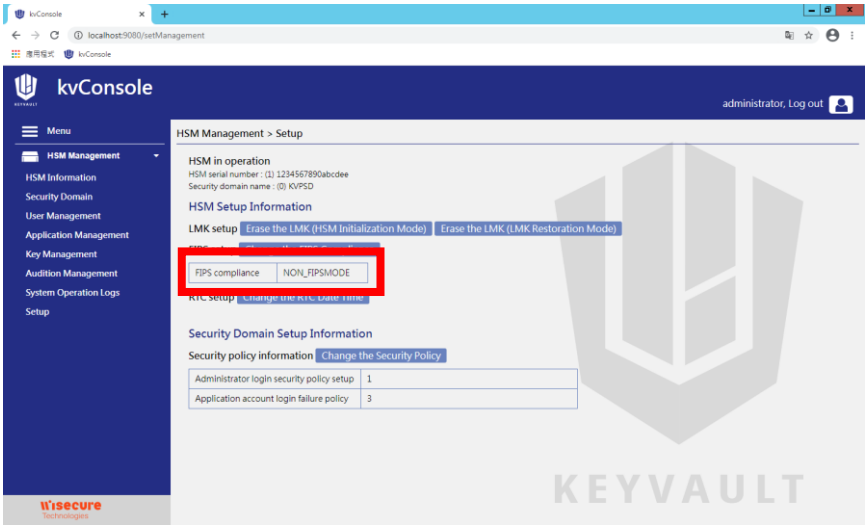


Figure 6 – Check FIPS mode

Administrator can change the mode by clicking the “Change the FIPS compliance” button and follow the steps.

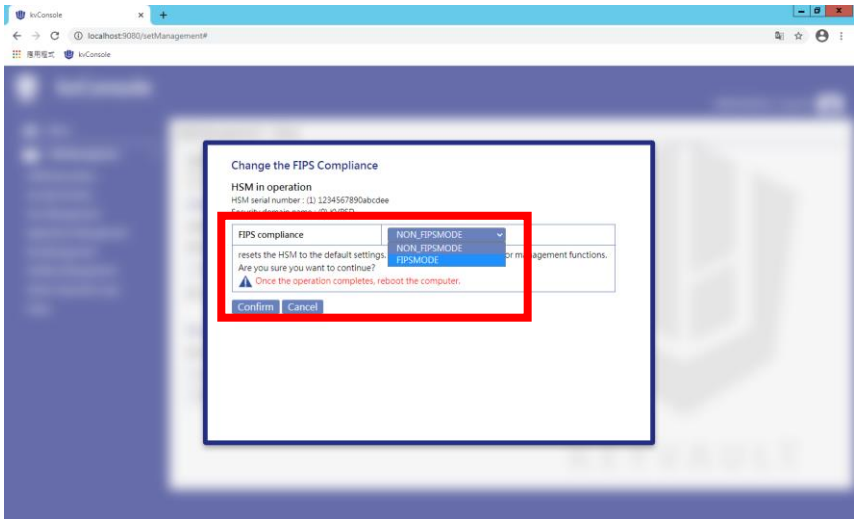


Figure 7 – Set FIPS mode

When a new mode of operation is selected by the operator, the module zeroizes the LMK, erases other encrypted CSPs, and requires a new initialization. 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.

1.4. Module Information

- Vendor: WiSECURE Technologies
- Product Name: kvHSM

- Module Name: KeyVault Hardware Security Module (kvHSM)
- Hardware Version: KV-HSM_V02
- Firmware Version: v1.00.0000

2. Cryptographic Module Ports and Interfaces

2.1. Physical Ports

Figure 8 shows the physical ports of the module. The module provides PCIe Gen2x8 physical interface. It provides the power source for the module and communication channel with the PCIe host. The USB host can be used to access the LMK token for key backup and restore. An RGB Status LED light indicates activity and status information of the module. The Erase Button can trigger the module to zeroize or restore the factory default configuration.

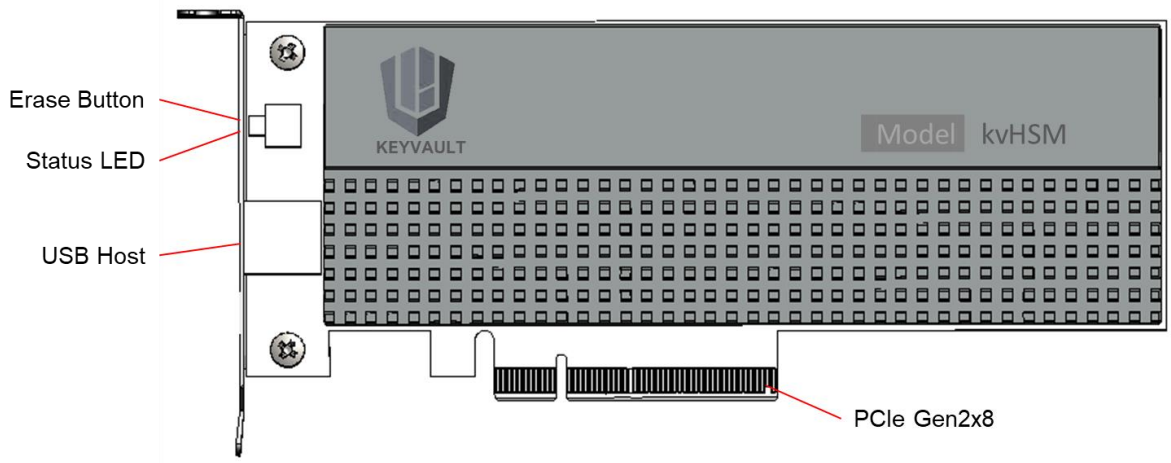


Figure 8 – Physical ports

The table below summarizes the physical ports.

Physical Ports	Pins Used	FIPS 140-2 Designation	Name and Description
USB Host Interface	USB Interface USB0_DP, USB0_DM	Data Output Data Input	The USB can also be used to access the LMK token for key backup and restore.
PCIe Gen2x8 Interface	PCIe Interface Lane 0 Transmit Side B (14, 15) Receive Side A (16, 17) Lane 1 Transmit Side B (19, 20) Receive Side A (21, 22) Lane 2 Transmit Side B (23, 24) Receive Side A (25, 26) Lane 3 Transmit Side B (27, 28) Receive Side A (29, 30)	Data Input Control Input Data Output Status Output Power	The module provide PCIe Gen2x8 physical interface. This interface provides the power source for the module and communication with the host.

Physical Ports	Pins Used	FIPS 140-2 Designation	Name and Description
	Lane 4 Transmit Side B (33, 34) Receive Side A (35, 36) Lane 5 Transmit Side B (37, 38) Receive Side A (39, 40) Lane 6 Transmit Side B (41, 42) Receive Side A (43, 44) Lane 7 Transmit Side B (45, 46) Receive Side A (47, 48)		
Status LED	RGB LED Interface	Status Output	The R-G-B status LED can indicate the status of the module, including Self-Testing (Green), Initialize (Green), Normal Status (Blue) and Error (Red).
Erase Button	Erase pin GPIO	Control Input	The Erase Button can trigger the board to zeroize the battery-backup SRAM credential and proceed the parameter erasing process when power is supplied. Long-press of the button will cause the module to restore the factory default configuration.

Table 2 – Physical ports

2.2. Logical Interfaces

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

Logical Interfaces	Physical Ports	Description
Data Input	USB Host	LMK restore with LMK token
	PCIe	Plaintext data from the host
	PCIe	Ciphertext or signed data from the host
	PCIe	Imported keys from the host
Data Output	USB Host	LMK backup with LMK token
	PCIe	Ciphertext or digital signatures to the host
	PCIe	Exported keys to the host
	PCIe	Management data to the host

Logical Interfaces	Physical Ports	Description
	PCIe	Device or control information to the host
Control Input	Erase Button	Trigger Zeroization
	PCIe	Management and Zeroization command from the host
Status Output	Status LED	Module status
	PCIe	Status request from the host
Power Input	PCIe	N/A

Table 3 – Logical Interfaces and their mapping with physical ports

Figure 9 indicates the logical interfaces mapping with physical ports.

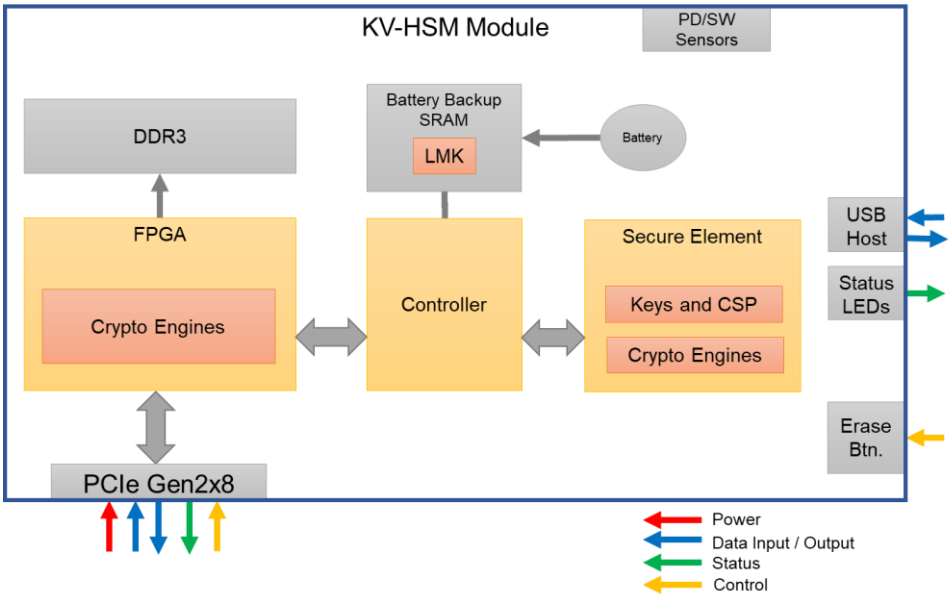


Figure 9 – Logical interfaces

3. Roles, Services and Authentication

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

The module 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 module verifies that the user has the proper role to perform the service.

3.1. Roles

The module supports 4 roles: Administrator, Operator, Auditor and Application. The first 3 roles are crypto-officer roles and can be assigned to users of the module (user) that perform management operations. The Application role is a user role assigned to external entities (application) that connect to the module through the middleware request cryptographic services.

A user can be assigned to only one fixed role: Administrator, Operator or Auditor. After the user is identified and authenticated to the module, the associated role is automatically assigned to the user.

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

The Application role is adopted implicitly by the external entities (applications) after they are identified and authenticated by 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	FIPS Roles
Administrator (ADM)	This role is assigned when a user with this role is authenticated successfully by 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.	Crypto Officer
Operator (OP)	This role is assigned when a user with this role is authenticated successfully by the module. The module supports only one user with this role. The operator role (OP) can perform system management and key management.	Crypto Officer
Auditor (AUD)	This role is assigned when a user with this role is authenticated successfully by the module. The module supports only one user with this role. The auditor role (AUD)	Crypto Officer

Role	Description	FIPS Roles
	can perform only audit management services (view management logs).	
Application (APP)	This role is assigned when an external entity (application) is authenticated successfully by the module via middleware. The Application role can request cryptographic services to the module.	User

Table 4 – Roles

3.2. Concurrent Users

The module supports concurrent users (including administrators and applications). The user requests session from the module and login with the session. The module maintains the session during the operation and close the session when user logout or timeout.

Each login session is bounded to the user and is limited to the functions and keys permit to the user.

3.3. Services

The module provides two types of services: management services and cryptographic server services. Management services include all services that can be executed from the management program by users of the module with the Administrator, Operator and Auditor roles. These services are classified into the following groups:

- Module Management
- Authority Management
- Key Management
- Backup and Recovery
- Application Management

Cryptographic server services are provided to applications that authenticates to the module through middleware.

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

3.3.1. Services in FIPS Mode of Operation

Table 5 below shows the functions 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	Description	Roles					Key/CSP	Access
		N/A	ADM	OP	AUD	APP		
Card Management								
Get HSM SN	Get HSM SN	*	*	*	*			
Get HSM Publickey	Get HSM Publickey	*	*	*	*			
Get Default CO Publickey	Get Default CO Publickey	*	*	*	*			
Get FIPS mode	Get FIPS mode	*	*	*	*			
Set FIPS mode	Set the FIPS mode or Non-FIPS mode		*	*			All Keys/CSPs	Delete
Erase LMK	Erase the LMK		*	*			LMK	Delete
Self-Test	Run self-test process	*	*	*				
SetTime	Set time		*	*				
Get HSM status	Get HSM overall status	*	*	*	*			
LMK and PSD Management								
Generate LMK	Generate the LMK		*	*			LMK	Write
Recover LMK	Restore the LMK		*	*			LMK	Write
Add PSD	Create the PSD of the module		*	*			SDMK SDMACK	Write Write
Init PSD SP	Init the security policy of the PSD		*	*				
Add PSD Admin	Add the admin to the security domain by PSD admin		*				ADM’s Public Key	Write
Session and Login/Logout								
Open Session	Open a login session	*	*	*	*	*		
Close Session	Close a login session		*	*	*	*		
Query Session	Query Session status		*	*	*	*		
Get Challenge	Get challenge to login		*	*	*	*		
CoToken Login	Login by CO Token, signed the challenge by token private key		*				CO’s Public Key	Read
CoPw Login	Login by CO password			*	*		CO Password	Read
UserPw Login	Login by Application password					*	Application Password	Read

Logout	Logout current user		*	*	*	*		
Security Domain Management								
List SD	Get the list of security domains	*	*	*				
Query SD Info	Query the SD Information	*	*	*				
Init SD Security Policy	Set the security policy of the security domain including minimum admin required to login and maximum pin trial before lock		*					
Update SD Security Policy	Update the security policy of the security domain.		*					
Add SD	Create a security domain by PSD admin		*				SDMK SDMACK	Write
Remove SD	Remove a security domain by PSD admin		*				SDMK SDMACK	Write
Backup SD	Backup a security domain by PSD admin		*				LMK	Read
Restore SD	Restore a security domain by PSD admin		*				LMK	Read
Administrator Management								
Add Admin	Add other admin(s)		*				New Admin's Public Key	Write
Remove Admin	Remove an admin		*				New Admin's Public Key	Write
List Admin	Get admin list		*					
Query Admin Info	Get admin info		*				Admin's Public Key	Read
Add Operator (Token-Based)	Add token-based operator		*				OP's Public Key	Write
Add Operator (ID/Password)	Add id-password based operator		*				OP's Password	Write

Remove Operator	Remove operator		*				OP's Public Key and Password	Write
Set Operator Password	Set operator's password by admin		*				OP's Password	Write
Change Operator Password	Change operator's password by operator			*			OP's Password	Write
List Operator	Get operator List		*					
Query Operator Info	Get operator Info		*	*				
Add Auditor (Token-Based)	Add token-based auditor		*				AUD's Public Key	Write
Add Auditor (ID/Password)	Add id-password based auditor		*				AUD's Password	Write
Remove Auditor	Remove auditor		*				AUD's Public Key and Password	Write
Set Auditor Password	Set auditor's password by admin		*				AUD's Password	Write
Change Auditor Password	Change auditor's password by auditor				*		AUD's Password	Write
List Auditor	Get auditor List		*					
Query Auditor Info	Get auditor Info		*		*			
Application Management								
Add Application	Add application		*				Application's Password	Write
Remove Application	Remove application		*				Application's Password	Write
Set Application Password	Set application password by admin		*				Application's Password	Write
Change Application Password	Change application password by application					*	Application's Password	Write
List Application	Get application List		*					
Query Application Info	Get application Info		*			*		
Key Management								
Gen Key	Generate keys for user		*	*			User Key	Write
Remove Key	Remove a key		*	*			User Key	Write
List Key	Get key List		*	*			User Key	Read

Query Key Info	Get key Info		*	*		*	User Key	Read
Update Key Status	Update the key status: active / inactive		*	*			User Key	Write
Update Key User	Update the user of the key		*	*			User Key	Write
Import Key	Import key from an encrypted keyblob		*	*			User Key TK	Write Read
Export Key	Export key as an encrypted keyblob		*	*			User Key TK	Read Read
Backup Key	Backup key as an encrypted keyblob		*	*			User Key SDMK	Read
Restore Key	Restore key from an encrypted keyblob		*	*			User Key SDMK	Write Read
Output Key (Init/Update/Final)	Split the key into shares and output		*	*			User Key	Read
Entry Key (Init/Update/Final)	Restore the key from shares		*	*			User Key	Write
Log Management								
Query Log Info	Query Log Info		*		*			

Table 5 – Management services in FIPS mode of operation

Table below shows the cryptographic services that can be requested in FIPS mode of operation, the involved cryptographic algorithms and the access required to keys and CSPs.

All services, except for “Application Authentication”, require the Application role in order to be executed. This role is obtained after the external entity authenticates successfully to the module (using the “Application Authentication” service).

Service	Algorithms	Key/CSP	Access
Authentication Service			
Application Authentication	DRBG, SHA2-256	Application Password	Read
Symmetric Key Services			
Key Generation	AES	AES Key	Create
Data Encryption	AES in ECB, CBC, OFB, CFB 126, CTR, modes	AES Key	Read
Data Decryption	AES in ECB, CBC, OFB, CFB 126, CTR modes	AES Key	Read
AES GCM Data Encryption	AES in GCM mod	AES Key	Read
AES GCM Data Decryption	AES in GCM mod	AES Key	Read

AES XTS Data Encryption	AES in XTS mod	AES Key	Read
AES XTS Data Decryption	AES in XTS mod	AES Key	Read
HMAC Generation	HMAC with SHA2-256, SHA2-384, SHA2-512, SHA3-256, SHA3-384, SHA3-512	HMAC Key	Read
Asymmetric Key Services			
ECDSA Key Pair Generation	DRBG, ECDSA	ECDSA Private Key	Create
ECDSA Public Key Export	ECDSA	None	N/A
ECDSA Digital Signature Generation	ECDSA	ECDSA Private Key	Read
ECDSA Digital Signature Verification	ECDSA	None	N/A
RSA Public Key Export	RSA 2048, RSA 4096	None	N/A
RSA Digital Signature Generation	RSA 2048, RSA 4096	RSA Private Key	Read
RSA Digital Signature Verification	RSA 2048, RSA 4096	None	N/A
Other Services			
Derive Key	EC Diffie-Hellman	ECC Private Key	Read
Message Digest	SHA2-256, SHA2-384, SHA2-512, SHA3-256, SHA3-384, SHA3-512 SHAKE128, SHAKE256	None	N/A
Random Number Generation	Hash_DRBG with SHA-256	Entropy Input String, Internal State	Read Update
Get FIPS status		None	N/A
Get HSM status		None	N/A

Table 6 – Cryptographic services in FIPS mode of operation

3.3.2. Services in Non-FIPS Mode of Operation

Table below shows the cryptographic 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 Application role in order to be executed. This role is obtained after the external entity is authenticated successfully by the module (using the "Application Authentication" service).

Service	Standard	Key Len / Curves	Usage
ECDSA Secp256k1	[ECCSEC2]	Secp256k1	Digital signature
ECC brainpool	[ECCBP]	P-256	Digital signature

Table 7 – Cryptographic services in Non-FIPS mode of operation

3.4. Algorithms

The module implements cryptographic algorithms in two separated components:

- WiSECURE Cryptographic Library (version 1.0), a firmware component that implements general purpose cryptographic algorithms used for cryptographic services and internal usage (AES for key protection and ECC Signature Verification for user authentication). CAVP: C1435 and A665.
- WiSECURE Cryptographic Hardware-Accelerated Library (version 1.0), an FPGA component that implements algorithms for high-speed cryptographic services. CAVP: C1706 and A1169.

3.4.1. Approved Algorithms

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.

CAVP Cert	Algorithm	Standard	Mode/Method	Key Len / Curves	Usage
C1435	AES	[FIPS 197] [SP 800-38A]	ECB, CBC, OFB, CFB 128, CTR	128, 192, 256	Data Encryption
C1435	AES-GCM	[SP 800-38D]	GCM	256	Data Encryption
C1435	AES-XTS	[SP 800-38E]	XTS	256	Data Encryption, Only used for storage
C1435	DRBG	[SP 800-90A]	Hash DRBG (SHA-256)	256	Random Number Generation
N/A	ENT (P)	[SP 800-90B]			Provide entropy input to the approved DRBG.
C1435	ECDSA KeyGen	[FIPS 186-4]		P-256/P-384/P-521	Key Generation
C1435	ECDSA KeyVer	[FIPS 186-4]		P-256/P-384/P-521	Key Verification
C1435	ECDSA SigGen	[FIPS 186-4]	SHA2-256/384/512	P-256/P-384/P-521	Signature Generation
C1435	ECDSA SigVer	[FIPS 186-4]	SHA2-256/384/512	P-256/P-384/P-521	Signature Verification
C1435	HMAC	[FIPS 198-1]		SHA2-256/384/512 SHA3-256/384/512	Message Authentication
C1435	SHS	[FIPS 180-4]		256/384/512	Message Digest
C1435	SHA-3	[FIPS 202]		224/256/384/512	Message Digest

				SHAKE128/256	
C1706	AES	[FIPS 197]	ECB, CTR	256	Data Encryption
C1706	AES	[SP 800-38E]	XTS	256	Data Encryption, Only used for storage
C1706	ECDSA KeyGen	[FIPS 186-4]		P-256	Key Generation
C1706	ECDSA SigGen	[FIPS 186-4]	SHA2-256	P-256	Signature Generation
C1706	ECDSA SigVer	[FIPS 186-4]	SHA2-256	P-256	Signature Verification
C1706	RSA KeyGen	[FIPS 186-4]		2048	Key Generation
C1706	RSA SigGen	[FIPS 186-4]	X9.31/PKCS#1.5/PSS SHA2-256	2048	Signature Generation
C1706	RSA SigVer	[FIPS 186-2]	X9.31/PKCS#1.5/PSS SHA2-256	4096	Signature Generation
C1706	RSA SigVer	[FIPS 186-4]	X9.31/PKCS#1.5/PSS SHA2-256	2048	Signature Verification
A665	AES-GCM	[SP 800-38D]		128, 192, 256	Data Encryption
A665	AES-XTS	[SP 800-38E]		128, 256	Data Encryption. Only used for storage
A665	KAS-ECC-SSC	[SP 800-56A]	(Cofactor) Ephemeral Unified Model, C(2e, 0s, ECC CDH) scheme	P-256/P-384/P-521	Key Agreement
A665	KDA	[SP 800-56C]	HKDF	256	Key Agreement
N/A	KAS		KAS-ECC-SSC Cert. #A665 and KDA Cert. #A665		Key Agreement
N/A	KTS		AES Cert. #C1435 and HMAC Cert. #C1435; key establishment methodology provides 256 bits of encryption strength.		Key Transport
A665	KBKDF	[SP 800-108]	Counter	256/384/512	Key Derivation
			Feedback	256/384/512	Key Derivation
			Double Pipeline Iteration	256/384/512	Key Derivation
A1169	RSA KeyGen	[FIPS 186-4]	SHA2-256	4096	Key Generation
A1169	RSA SigGen	[FIPS 186-4]	SHA2-256	4096	Signature Generation
Vendor	CKG	[SP 800-133]	● Key Pairs generation using unmodified DRBG output for Digital		

Affirmed			Signature Schemes ● Key Pairs generation using unmodified DRBG output for Key Establishment ● The “Direct Generation” of Symmetric Keys generation using unmodified DRBG output ● Symmetric Keys Generated Using Key-Agreement Schemes ● Distributing Symmetric Keys using key wrapping

Table 8 – Approved algorithms

There are algorithms, modes, and keys that have been CAVP tested but not used by the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by the module.

Some algorithms (AES, RSA and ECDSA) has FPGA implementations that provide high-speed cryptographic services. Those services can be used with imported wrapped keys and session keys derived from key agreement service.

For AES GCM encryption, the IV is generated internally by the module’s Approved DRBG. The DRBG seed is generated inside the module’s physical boundary. The IV is 96 bits in length per [SP 800-38D], Section 8.2.2 and FIPS 140-2 [IG] A.5 scenario 2.

In case the module’s power is lost and then restored, the key used for AES GCM encryption or decryption shall be re-distributed.

When a GCM IV is used for decryption, the responsibility for the IV generation lies with the party that performs the AES-GCM encryption therefore there is no restriction on the IV generation.

3.4.2. Non-Approved Algorithms

The table below shows the algorithms that are not approved in FIPS mode of operation.

Algorithm	Standard	Usage
ECDSA	[ECCSEC2]: Secp256k1	Digital signature
ECDSA	[ECCBP]: Brainpool 256	Digital signature

Table 9 – Non-Approved algorithms

3.5. Identification and Authentication

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

- Token-based authentication: For Administrator, Operator or Auditor roles, it can apply the Token

-based authentication. It is performed using a challenge-response mechanism with the user's credentials (a 256-bit ECC key pair) stored in the user's USB token.

- ID/Password authentication: For Operator, Auditor and Application roles, it can apply the ID/Password-based authentication. It verifies the user's credential stored in the module.

Roles	FIPS Roles	Type of Authentication	Authentication Data
Administrator	Crypto Officer	Token-Based	Public and Private Key
Operator with Token	Crypto Officer	Token-Based	Public and Private Key
Auditor with Token	Crypto Officer	Token-Based	Public and Private Key
Operator with ID/Password	Crypto Officer	ID/Password	ID/Password
Auditor with ID/Password	Crypto Officer	ID/Password	ID/Password
Application	User	ID/Password	ID/Password

Table 10 – Roles and authentication mechanisms

3.5.1. Token-Based Authentication

When a new user of the module is added, the module then asks for the insertion of the new user's USB token, imports and stores the user's ECC public key together with the ID and the associated role of the user.

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

1. The user asked for token-based login through kvConsole
2. The module generates a challenge consisting of a 256-bit random number
3. The user inserts the Admin Token into the USB port, and enters the PIN through the kvConsole
4. The Admin Token authenticates with the PIN provided by the user.
5. The Admin Token generates a digital signature of the challenge as response. (using the user's ECC private key stored in the Admin Token)
6. The kvConsole ask for further tokens to process step 3-5 according to the security policy.
7. The kvConsole sends the response(s) to the module for login
8. The module verifies the signature(s) with the users' public key(s) (stored in the module)

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

3.5.2. ID/Password Authentication

Administrator can set the Operator and Auditor Role with ID and password to the module. The module identifies and authenticates users of the module through the following steps:

1. User enters the ID and Password through the kvConsole.
2. The kvConsole gets a challenge from module
3. The kvConsole calculate the user ECC private key from password (SHA256(password))
4. The kvConsole calculate the response as ECDSA of the SHA256(challenge) with user ECC private key
5. The kvConsole send the user ID and response to module to login
6. The module verifies the response by user's public key, which is calculate from user's password stored inside the module.

If the signature verified success, then authentication succeeds, the user adopts the role assigned during its creation. 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).

3.6. Authentication Strength

There are 2 types of user authentication mechanism.

- Token-based authentication:
  The Token-based authentication mechanism can be applied to administrator, operator and auditor. It uses a 256-bit ECC key pair. According to [SP 800-57A], such a key provides a security strength of 128 bits. Therefore, the probability of a successful authentication by guessing the private key, using a USB token with a non-registered user's credential, is $2^{-128} \approx 10^{-38}$, which is far less than the maximum probability of 10^{-6} required by the FIPS 140-2 standard.

  For the token-based authentication it requires user to plug in token manually. It assumes user can perform a maximum 100 attempts in one minute, the total probability of guessing the credentials is $2.93 \times 10^{-39} \times 100 = 2.93 * 10^{-37}$. This number is still far less than one in 100,000.

- ID/Password authentication:
  The ID/Password authentication mechanism can be applied to operator, auditor and user application. It requires the length of password to be 8 characters at least, and it limits the maximum trial time of authentication. The module will locked if the fail-trial exceed the maximum trials and it will require the administrator to unlock for further usage.

  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.

  For the ID/Password authentication, the Module will be locked if password fail-trial exceed the maximum trial, say 255 times. In this case the successful random attempt during one minute is

$3.54 \times 10^{-13} \times 255 = 9.03 \times 10^{-11}$, which is less than one in 100,000.

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

The module is covered with an aluminum cover. It covers all the major components and tamper response circuits that can detect the event of the disclosure of enclosure. On the bottom side of the PCB a metal frame is attached directly onto the printed circuit board, and the space inside the metal frame is filled by hard and opaque potting material which show evidence of tampering on the enclosure when a physical attack is attempted.

One tamper-evident seal is pre-installed (at factory) as shown on the Figure 10 with red circle. The tamper-evident seal covers joint seam and a screw on the middle of back side. The aluminum cover cannot be removed or displaced without removing the screw covered by the tamper-evident seal. The tamper-evident seal cannot be penetrated or removed and reapplied without evidence of tampering. The tamper-evident seal shall be installed for the module to operate in the Approved mode of operation.

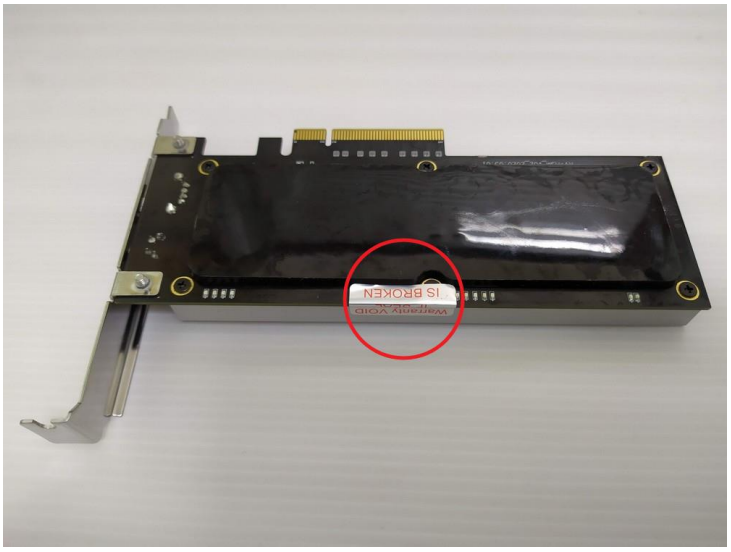


Figure 10 – Tamper-evident seal

4.2. Dynamic Protection

During the operation of the module, keys and CSPs are stored in plaintext into the volatile memory. In order to prevent the disclosure of such sensitive information, the module with tamper-evident

enclosure (the heat sink and the potting material) implements the following physical security mechanisms:

- The module implements a tamper response and zeroization circuitry which will be activated while the cover is removed.
- The cryptographic module's hardware components are covered by hard, opaque potting material or the heat sink which show evidence of tampering on the enclosure when a physical attack is attempted.
- The potting material is hard and opaque enough to prevent direct observation and easy penetration to the depth of the underlying hardware components. It is highly probable that anyone attempting to penetrate to the depth of the circuitry will break off large pieces of potting material and tear important hardware components off the module, causing serious damage to the module.

Users should check the tamper-evident seal and status LED periodically to maintain the physical security.

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	Key Size / Curve	Purpose	Generation	Entry and Output	Storage	Zeroization
LMK AES-256 Key	256 bits	Protect each SD MK	During initialization, [SP 800-133]. (unmodified output from the approved DRBG)	Input from LMK Tokens (split knowledge) Output to LMK tokens (split knowledge)	Battery-Backup SRAM	When Erase Event Triggered
SD MK AES-256 Key	256 bits	Protect each Keys and CSPs stored in the SD	During SD initialization, [SP 800-133]. (unmodified output from the approved DRBG)	Input and output as service requests (key wrapping)	SLE97 Flash Encrypted by LMK Protected by SD MAC key	Erased when SD deleted
SD MAC Key HMAC	256 bits	Protect each Keys and CSPs integrity	During SD initialization, KBKDF from SD MK	N/A	SLE97 Flash Encrypted by LMK	Erased when SD deleted
Administrator ECC Public Key	P-256	Identity-based Authenticati on	None (Generated by USB Token)	ECC Public Key is input from the USB token	SLE97 Flash In Plaintext Form Protected by SD MAC key	Erased when user deleted or initialized
Operator ECC Public Key	P-256	Identity-based authenticatio n	None (Generated by USB Token)	ECC Public Key is input from the USB token	SLE97 Flash In Plaintext Form Protected by SD MAC key	Erased when user deleted or initialized
Auditor ECC Public Key	P-256	Identity-based authenticatio n	None (Generated by USB Token)	ECC Public Key is input from the USB token	SLE97 Flash In Plaintext Form Protected	Erased when user deleted or initialized

Name	Key Size / Curve	Purpose	Generation	Entry and Output	Storage	Zeroization
					by SD MAC key	
Operator Password	8 bytes min.	Identity-based authentication	N/A	Input from kvConsole	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Erased when user delete or initialized
Auditor Password		Identity-based authentication	N/A	Input from kvConsole	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Erased when user delete or initialized
User AES Keys	256 bits	Encrypt/Decrypt data	Generated using [SP 800-133]. (unmodified output from the approved DRBG)	Input and output as service requests (wrapped key)	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Management Services
User KEK	256 bits	Encrypt/Decrypt user key	Generated using [SP 800-133]. (unmodified output from the approved DRBG)	Input from TK Tokens (split knowledge) Output to TK tokens (split knowledge)	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Management Services
User MAC Key	256 bits	Protect Wrapped Key Integrity	KBKDF from User KEK	N/A	N/A (in RAM only)	Erased when no longer used.
User ECDSA Public Keys	P-256 P-384 P-521	Digital Signature Verification	Generated from ECDSA private key	Input and output as service requests	SLE97 Flash Protect by SD MAC key	Management Services
User ECDSA Private Keys	P-256 P-384 P-521	Digital Signature Generation	Generated using [SP 800-133]. (unmodified output from	Input and output as service requests (wrapped key)	SLE97 Flash Encrypted by SD MK Protected by SD MAC	Management Services

Name	Key Size / Curve	Purpose	Generation	Entry and Output	Storage	Zeroization
			the approved DRBG)		key	
User EC Diffie-Hellman Public Keys	P-256 P-384 P-521	Key Agreement	Generated from EC Diffie-Hellman private key	Input and output as service requests	SLE97 Flash Protect by SD MAC key	Management Services
User EC Diffie-Hellman Private Keys	P-256 P-384 P-521	Key Agreement	Generated using [SP 800-133]. (unmodified output from the approved DRBG)	Input and output as service requests (wrapped key)	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Management Services
EC Diffie-Hellman Share Secret	256 bits	Derive Secure Channel Session Key	Established during EC Diffie-Hellman Exchange	N/A	N/A (in RAM only)	Erased when no longer used.
Secure Channel Session Key	256 bits	Protect Channel Payload	KAS from EC Diffie-Hellman Share Secret	N/A	N/A (in RAM only)	Erased when no longer used.
User RSA Public Keys	2048 bits 4096 bits	Digital Signature Verification	N/A	Input and output as service requests	SLE97 Flash Protect by SD MAC key	Management Services
User RSA Private Keys	2048 bits 4096 bits	Digital Signature Generation	N/A	Input and output as service requests (wrapped key)	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Management Services
Application Password	8 bytes min	Identity-based Authentication of Application	N/A	Input by user with Administrator from kvConsole.	SLE97 Flash Encrypted by SD MK Protected by SD MAC key	Erased when user delete or initialized
Entropy Input		Entropy for DRBG	Obtained from ENT (P)	None	N/A (in RAM only)	Erased when no longer used.
DRBG		DRBG	During DRBG	N/A	N/A (in	Erased when

Name	Key Size / Curve	Purpose	Generation	Entry and Output	Storage	Zeroization
Internal State (V&C)		Internal State	Initialization		RAM only)	DRBG is no longer used.

Table 11 – Lifecycle of keys and CSPs

6.1. Random Number Generation

The module employs a Deterministic Random Bit Generator (DRBG) compliant to [SP 800-90A] for the creation of symmetric and asymmetric keys, creation of random number challenges for the identity-based authentication mechanism, and processing of the Random Number Generation service request.

The DRBG supports the Hash_DRBG mechanisms. The DRBG is initialized during module initialization; the module loads by default the DRBG using the Hash_DRBG mechanism with SHA-256.

For seeding the DRBG, the module uses a non-deterministic Random Number Generator (NDRNG) which is a SP 800-90B compliant physical entropy source, i.e., ENT (P). The ENT (P) is provided by FPGA implementation within the boundary. The ENT (P) provides around 0.64 bits entropy per byte. During initialization (seed) and reseeding (reseed) process, we feed DRBG by 768 bits of ENT (P) as entropy input; more specifically, using the terminology from [SP 800-90A], the 768 bits of ENT (P) is divided into 512 bits of entropy input and 256 bits of nonce of Hash_DRBG, with personalization string being empty. Since the ENT (P) provides around 0.64 bits entropy per byte, the 768 bits entropy input contains larger entropy than 256 bits, that is, larger than the security strength of Hash_DRBG SHA-256 algorithm (0.64 x 512 > 256).

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.

AES symmetric keys are obtained using unmodified output from the approved DRBG. Keys are generated using random data from the [SP 800-90A] DRBG following the [SP 800-133]. ECDSA key pairs are generated in compliance with [FIPS186-4] and using the [SP 800-90A] DRBG and follow the [SP 800-133].

The LMK is an AES symmetric key and the value is split into 3 key components (shares) according to Shamir's Secret Sharing with threshold 2. Each share is stored in LMK token and any 2 of the 3 shares can reconstruct the original LMK.

The [SP 800-108] KBKDF is used to generate SD MAC Key, User MAC key, Secure Channel Session Key, and for key derivation service.

6.3. Key Agreement (Establishment)

The module provides the following key agreement scheme to establish a key between two parties. Each party contribute info to construct the key.

- EC Diffie-Hellman (key agreement; key establishment methodology provides 128 / 192 / 256 bits of encryption strength). It utilizes ECC with Curve P-256 / P-384 / P-521 for the 128 / 192 / 256 bits AES key agreement respectively. EC Diffie-Hellman is applicable for which scenario X1 (2) of [IG] D.8 Key Agreement Methods. The session key is derived by HKDF key derivation function documented in Section 5.1 of [SP 800-56C].

The usage of the key agreement includes:

- Generate secret session key of the trusted path between the API and the module.
- Provide key agreement service.

6.4. Key Transport (Establishment)

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 CBC mode and HMAC-SHA-256 algorithms. The certificate is KTS (AES Cert. #C1435 and HMAC Cert. #C1435; key establishment methodology provides 256 bits of encryption strength).

According to “Table 2: Comparable strengths” in [SP 800-57], the key sizes of AES provide the following security strength:

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

6.5. Key Entry and Output

The module supports electronic distribution of keys in encrypted form. The module does not enter or output keys in plaintext format outside its physical boundary.

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-SHA-256

as the key transport mechanism, using one of the keys stored in the module.

LMK can be input from or output to external LMK tokens through the key management services. The keys are input or output in split key component form based on Shamir's Secret Sharing algorithm.

6.6. Split Knowledge Procedure

The module uses the split knowledge procedure for entry and output of the Local Master Key (LMK) and Transport Key (TK). 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 reconstruct the original key when they are entered the module.

6.7. Key / CSP Storage

All keys and CSPs are stored in encrypted form into the non-volatile memory. The module protects keys and CSPs using the same mechanism used for key wrapping (AES in CBC mode and HMAC-SHA-256) compliant with [SP 800-38F]. 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 Local Master Key (LMK), which is stored in the battery-backup SRAM.

6.8. Key / CSP Zeroization

The zeroization process is triggered by the operator, tamper response and switch FIPS mode.

When the operator sends a zeroization host command or press erase button, it will perform the following processes in order:

1. Zeroize LMK stored in battery-backup SRAM
2. Erase Encrypted Keys and CSPs stored in Flash

It will zeroize the LMK by writing 0x00s to overwrite the key resident in battery-backup SRAM. And all the Encrypted Keys and CSPs stored in Flash will be erased. The erase command must be signed by an authorized administrator or operator, or it must press the erase button for 10 seconds.

When the module detects a tamper event of opening the enclosure, the zeroization circuit will cut the internal battery power and cause the SRAM shutdown and lose the LMK.

The LMK resident in the SRAM can be erased within 2ms when the zeroization process is triggered. The time is not enough for the attacker to compromise the plaintext secret and private keys.

7. 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.

8. Self-Tests

The module performs the following self-tests: Integrity Test, Continuous Random Generator Test, self-tests for a set of critical functions. All critical functions are listed in Table 12 and Table 13. The functions in Table 12 are tested immediately after power-up, and functions in Table 13 are tested conditionally.

8.1. Power-Up Self-Tests

Power-up self-tests include integrity tests on the firmware and cryptographic algorithm self-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 are executed automatically when the module powered on. While the module is executing self-tests, input and output are inhibited. Input and output are inhibited, services are not available until all self-tests complete successfully.

When the module finishes the power-up self-tests, the status LED indicates the status. The blink blue light indicates the test succeed and the module is ready for service. If any of the power-up self-test fails, the module enters the error state and it will blink red light to indicate an error status. Input and output are inhibited and none of the management or cryptographic services is available. The operator can get the test log via management console.

8.1.1. Integrity Tests

The integrity of the module is verified by comparing a SHA-256 value calculated at run time with the value stored in the module. The SHA-256 value in the module is the hash value of firmware code and is computed during the module production process. During the integrity test, the firmware code is used again to compute a SHA-256 value, which is used to compare with the stored value. If the two SHA-256 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	● AES (encrypt/decrypt) KATs, 256-bit key for modes ECB/CBC/CFB128/CTR/GCM/OFB/XTS (Cert. #C1435, #A665) ● AES (encrypt/decrypt) KATs, 256-bit key for modes ECB/CTR/XTS (Cert. #C1706)
SHS	● SHA2 KATs, SHA2-256, 384, 512 (Cert. #C1435) ● SHA3 KATs, SHA3-224, 256, 384, 512 (Cert. #C1435) ● SHAKE KATs, SHAKE-128, 256 (Cert. #C1435)
HMAC	● HMAC KATs, HMAC-SHA2-256, 384, 512 (Cert #C1435) ● HMAC KATs, HMAC-SHA3-256, 384, 512 (Cert. #C1435)
ECDSA	● ECDSA PCT with P-256, P-384 and P-521 (Cert. #C1435) ● ECDSA PCT with P-256 (Cert. #C1706)
EC Diffie-Hellman	● KAS-SSC-ECC KATs, P-256 (Cert. #A665) ● Primitive Z computation KAT (Cert. #A665) (per [SP 800-56A], Section 5.7.1.2). ● KDF KAT (Cert. #A665) (per [SP 800-56A], Section 5.8.1).
RSA	● RSA 2048 and 4096 PCT (Cert. #C1706 and #A1169)
KBKDF	● KBKDF KATs, 256 (Cert. #A665) (per [SP 800-108])
DRBG	● KAT Hash_DRBG using SHA-256 (Cert. #C1435)

Table 12 – Power-Up self-tests

For KATs, the module calculates the result and compare 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

The operator can issue the management command or restart the module to trigger the power-up self-tests.

8.3. Conditional Self-Tests

The module performs conditional tests on the cryptographic algorithms as shown in the following table.

Algorithm	**Test**
ECDSA	Pair-wise Consistency Test
RSA	Pair-wise Consistency Test
ENT (P)	Continuous Random Number Generator Test (CRNGT) Health Test compliant with SP 800-90B: Repetition Count Test (RCT) and Adaptive Proportion Test (APT)

DRBG	Continuous Random Number Generator Test (CRNGT) Health Test compliant with SP 800-90A including: KAT for the Instantiate, Generate and Reseed functions.

Table 13 – Conditional self-tests

If a conditional test fails, the module enters the error state. Input and output are inhibited and none of the management or cryptographic services is available.

The Continuous Random Number Generator Test (CRNGT) is performed on the ENT (P) and DRBG supported by the module. An initial random number is generated and stored upon power up for both ENT (P) and DRBG. A successive call to ENT (P)/DRBG generates a new set of data, which is compared to the comparison data. If a match is detected, this test fails; otherwise the new data is stored as the comparison data and returned to the caller. The module enters the error state if this test fails

9. Guidance

In 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 Session 3.3). In addition, cryptographic algorithms and their key sizes must also comply with [SP 800-131A].

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. Initialization

The module is shipped to the vendor without any initialization of keys or CSPs. By default, the module 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 module initialization without any authentication.
The Administrator must perform the initialization of the module following the instructions.

- 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.
- Open the package and verify with the content list that the module and all accessories are included.
- Verify the condition of the module and closure to see if there are signs of damage.
- Connect the module to a server PCIe slot.
- Power-up the server.
- Install and Run the module initial program (kvConsole).
- Verify that the product model and version provided in the “View Device Basic Information” menu option matches the following information:
  - Vendor: WiSECURE Technologies
  - Product Name: kvHSM
  - Module Name: KeyVault Hardware Security Module (kvHSM)
  - Hardware Versions: KV-HSM_V02
  - Firmware Versions: v1.00.0000
- Generate Admin Tokens
  - Use kvConsole to Initialize Admin Tokens (x3)
- Use the Installation Wizard to perform the following activities:
  - Initialize the device by Admin Tokens. (Set admin Public-keys into module)
  - Create the Local Master Key (LMK) and export the LMK key components to LMK Tokens.
  - Create the Operator and Auditor
  - Configure the Device security policies

9.2. USB Tokens

To initialize the module, the following USB Tokens must be available:

- 3 Admin Tokens for the device and user management.
- 3 LMK Tokens for exporting the LMK key components.

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

Operator and Auditor can use the Token as the login credential or set a password.

10. Mitigation of Other Attacks

The module has been designed to mitigate against power analysis attacks and physical probing, which are outside of the scope of FIPS 140-2.

Other Attack	Mitigation Mechanism
Side-channel attacks (power analysis)	The module hardware provides desynchronization and confusing mechanism to protect security calculation against SPA and DPA power analysis attacks. The module implements following mitigations against the power analysis attacks: Montgomery powering ladder, exponent blinding, scalar randomization, random projective coordinates, random register address, point validation, curve integrity check and Two-share masking.

Table 14 – Mitigation of other attacks

11. Security Rules

The module design corresponds to the module's security rules. This section documents the security rules enforced by the module to implement the Level 3 security requirements of FIPS 140-2.

1. The module supports two modes of operation, FIPS mode and non-FIPS mode. Power-cycling zeroizes all volatile plaintext critical security parameters.
2. The FIPS mode is implicitly assumed when an Administrator initializes the module for the first time.
3. No hardware, software, or firmware components of the cryptographic module are excluded from the security requirements of FIPS 140-2.
4. All data output via the data output interface are inhibited when an error state exists and during self-tests.
5. The output data paths are logically disconnected from the circuitry and processes that perform key generation, and key zeroization.
6. The module never outputs plaintext cryptographic keys or CSPs or sensitive data.
7. Status information never contains CSPs or sensitive data that if misused could lead to a compromise of the module.
8. The module supports maximum 16 concurrent operators.
9. The module does not support a maintenance role.
10. The module does not support a bypass capability.
11. Authentication data within the module is protected against unauthorized disclosure, modification, and substitution.
12. The module contains the authentication data required to authenticate the operator for the first time.
13. The module’s authentication mechanism does not supply any feedback information to the operator.
14. Recovery from “soft” error states is possible via power-cycling. Recovery from “hard” error states is not possible.
15. The operator should periodically inspect the module for evidence of tampering.
16. Secret keys, private keys, public keys, and CSPs within the module are protected from unauthorized disclosure, modification, and substitution.
17. Compromising the security of the key generation methods requires as least as many operations as determining the value of the generated keys.
18. Intermediate key generation values are not output from the module.
19. The module does not support manual key entry.
20. The module does not support a SW/FW Load service from the operator (host).

12. References

[ECCBP] RFC 5639: Elliptic Curve Cryptography ECC Brainpool Standard – Curves and Curve Generation

[FIPS 140-2] FIPS PUB 140-2 - Security Requirements for Cryptographic Modules

[FIPS 180-4] Secure Hash Standard (SHS)

[FIPS 186-2] Digital Signature Standard (DSS)

[FIPS 186-4] Digital Signature Standard (DSS)

[FIPS 197] Advanced Encryption Standard (AES)

[FIPS 198-1] The Keyed Hash Message Authentication Code (HMAC)

[FIPS 202] SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

[IG] Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program

[ECCSEC2] SEC2: Recommended Elliptic Curve Domain Parameters

[SP 800-38A] NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation: Methods and Techniques

[SP 800-38D] NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC

[SP 800-38E] NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices

[SP 800-38F] NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping

[SP 800-56A] NIST Special Publication 800-56A Revision 3 - Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography

[SP 800-56C] NIST Special Publication 800-56C Revision 2 - Recommendation for Key-Derivation Methods in Key-Establishment Schemes

[SP 800-90A] NIST Special Publication 800-90A Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators

[SP 800-108] NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions

[SP 800-131A] NIST Special Publication 800-131A Revision 2 - Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

[SP 800-133] NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation

13. Definitions and Acronyms

AES	Advanced Encryption Standard
AMK	Application Master Key
APP	Application
CC	Common Criteria
CRNGT	Continuous Random Number Generator Test
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DPA	Differential Power Analysis
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
HSM	Hardware Security Module
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDF	Key Derivation Function
KBKDF	Key-Based Key Derivation Function
HKDF	HMAC-based Extract-and-Expand Key Derivation Function
HMAC	Keyed Hash Message Authentication Code
LMK	Local Master Key
MAC	Message Authentication Code
MBK	Master Backup Key
NDRNG	Non-deterministic Random Number Generator
PCT	Pair-wise Consistency Test
PSD	Primary Security Domain
RNG	Random Number Generator
SCA	Side-channel Attack
SD	Security Domain
SHA	Secure Hash Algorithm
SP	Security Policy
SPA	Simple Power Analysis
SSC	Shared Secret Computation
TK	Transport Key