IBM 4769-001 Enterprise PKCS#11 HSM Cryptographic Coprocessor Security Module

FIPS 140-2 Non-Proprietary Security Policy

Document Version 1.4

Last update: 2023-07-18

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1 Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy of the IBM 4769-001 Enterprise PKCS#11 HSM Cryptographic Coprocessor Security Module. It describes the design and features of this cryptographic module and the rules under which the module operates. It also describes how the module meets the requirements of FIPS 140-2 for all applicable areas at security level 4.

1.1 Module Overview

The IBM 4769-001 Enterprise PKCS#11 HSM Cryptographic Coprocessor Security Module cryptographic module (hereafter referred to as “the module”) is a Multi-Chip Embedded Hardware cryptographic module in the form of a PCIe card. This module with components Miniboot (MB) hardware and EP11 firmware, provides crypto officers and users the security policy governing access to the services specified in section 3.2. EP11 provides an interface similar to the industry standard PKCS#11 API. The EP11 firmware provides a stateless backend, relying mainly on host-resident, encrypted datastores to maintain sensitive state, while presenting services as a regular HSM-based PKCS#11 implementation.

The Module is a cryptographic coprocessor, a general-purpose computing platform with cryptographic accelerator engines, executing firmware and retaining secrets, despite foreseeable physical or logical attacks. The overall security rating of the module is 4. The Module is intended for use by US Federal agencies and other markets that require FIPS 140-2 validated Level 4 modules. End users can base high-assurance applications, such as digital signature generation or financial transaction processing, on this platform. Table 1 lists the security levels supported by the cryptographic module according to each section of FIPS 140-2.

FIPS 140-2 Section Title	Security Level
Cryptographic Module Specification	4
Cryptographic Module Ports and Interfaces	4
Roles, Services, and Authentication	4
Finite State Model	4
Physical Security	4
Operational Environment	N/A
Cryptographic Key Management	4
EMI/EMC	4
Self-Tests	4
Design Assurance	4
Mitigation of Other Attacks	N/A
Overall	4

Table 1 - Security Levels

For the purposes of this FIPS 140–2 validation, this policy describes fixed module configurations, and does not allow firmware updates. Therefore, this policy is applicable only when the appropriate configurations are loaded to suitable hardware. Components are identified through the

most significant eight bits (i.e. the first byte) of their content hashes, which is reported by host drivers in a platform-specific way. The configurations covered by this policy are the following:

Model	Hardware [Part Number and Version]	Configuration
4769-001	PN 02WN654-N37880 POST0 v9662 MB0 v6096 (Standard Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701 Hash data: 2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932 217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data: 1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC 8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011
4769-001	PN 02WN652-N37880 POST0 v9662 MB0 v6096 (Low Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701 Hash data:

		2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data: 1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC 8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011
4769-001	PN 03FM956-H07053 POST0 v8657 MB0 v6381 (Standard Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701 Hash data: 2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932 217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data: 1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC

		8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011
4769-001	PN 03FM953-H07053 POST0 v8657 MB0 v6381 (Low Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701 Hash data: 2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932 217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data: 1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC 8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011
4769-001	PN 03JJ168-N38177 POST0 v8657 MB0 v6381 (Standard Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701

		Hash data: 2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data: 1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC 8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011
4769-001	PN 03JJ165-N38177 POST0 v8657 MB0 v6381 (Low Power)	Segment 0 Information The hardware part numbers imply Segment 0 configuration Segment 1 Information Name: 7.0.46z P1591 M1591 P5625 F0701 Hash data: 2B5F92F34C8FF2CDC93B794AE6F4EA44B4C5112A5FA9215 6F8E67BC1F1B4F557E9BC92F4CEE9C896C1F560D954F873 54E64F60BC28535765127CBE8985E07C06 Name: 7.0.74z P3795 M6356 P0630 F0701 Hash data: 5D4F8741EDD2403F61C33D3C190B714D5A3B421DD38E40 94547C3C3B229CC5217F94324B4840AB98EAE7644AD87E8 932 217CC15CBD045A83F33F8D48DC6E7AF6 Segment 2 Information short ID: 1ED5 682D (the first bits of the hash value of segment firmware) Hash data:

		1ED5 682D 630F 337D 8102 AF8A 0B0B C696 8281 CEFA 1174 01AF 2EA7 BA5B 9D7B 67D7 C5E0 10E9 FC3F 57BC 8397 8F5D 49A5 30AA ED6B 8231 9192 2988 583A BC7F 9F17 C035 Segment 3 Information short ID: 7C37 01B6 Hash data: 7C37 01B6 4CCF F8D1 66DE 81EF 41DB 793C EAC1 BDB2 4BFA A9E5 B315 6EF2 3AA7 A00D FAD9 87B1 2209 6AE6 770A 19F4 DF54 14ED 158D 70CE 1A78 6244 4D68 1084 A6AE A011

Table 2 - Cryptographic Module Components

Figure 1 and Figure 2 show representations that apply to all part numbers listed in Table 2. Although the part numbers in the table are different, the module functionality and the structure are the same for all 4769-001 with the part numbers listed in Table 2. The physical form of the 4769-001 PCIe module is depicted in Figure 1. The red outline in the picture depicts the physical cryptographic boundary. The module is comprised of two (2) electrical component cards with one used as a battery holder and the second one being the main functional component of the Module. The module relies on a host system that supplies a PCIe interface for input/output communication.

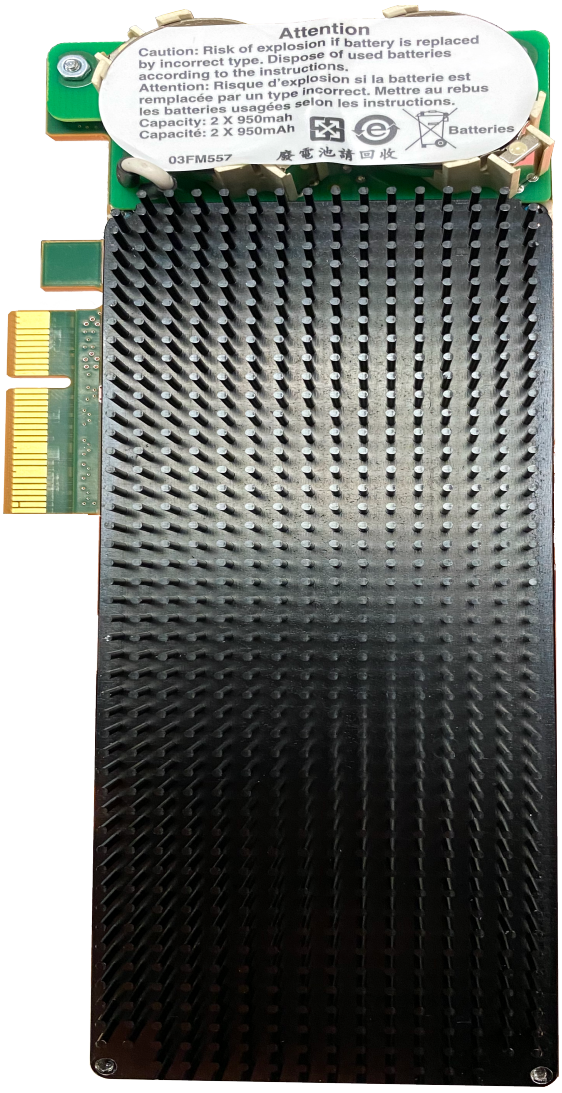

Figure 1 – Module

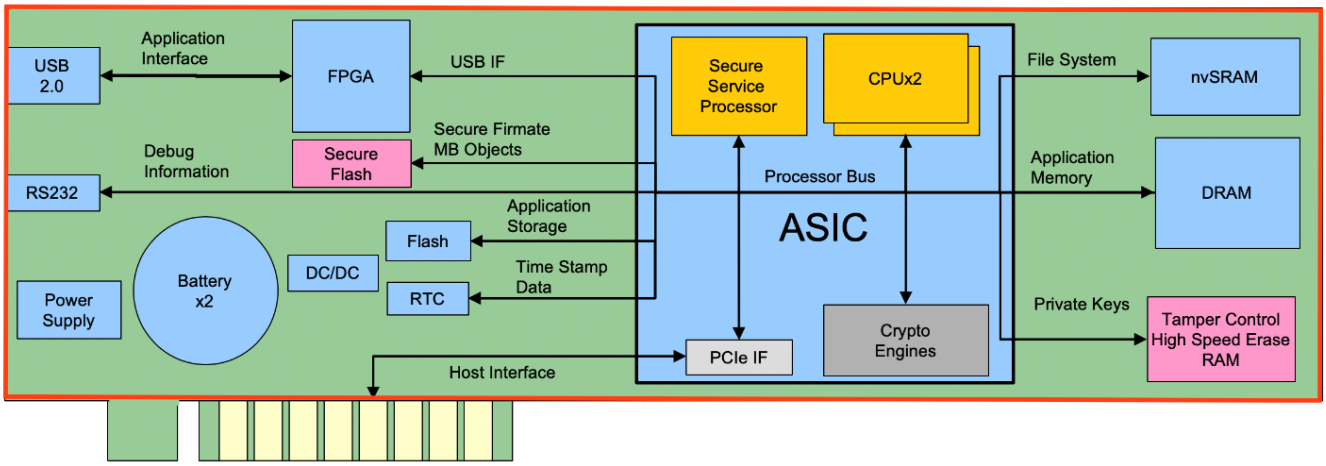

Figure 2 – 4769-001 Block Diagram

The module is divided into four layers, which are further referred to as segments, as illustrated in Figure 3. The items on the left side of the segments in Figure 3 are not part of the module. The base two segments, and a stub in the third segment control security and configuration of the module:

- Segment 0: Permanent POST 0 (Power-on Self-Test) and Miniboot 0 (security bootstrap). This code is in Secure Flash, bootstrapping the entire module, effectively non-modifiable.
- Segment 1: Rewritable POST 1 and Miniboot 1, responsible for most of user-visible infrastructure functionality.

  POST 2, while executed by the module CPU, is logically controlled and is considered as part of Segment 1. Specifically, POST 2 gets control immediately after module CPU reset and before any OS or higher-level applications. POST 2 does not get access to secrets, and it must be approved by the Segment 1 crypto officer to load (being part of Segment 1 firmware updates).

  POST routines perform initial and higher-level testing of the module's infrastructural functionality. If both POST 0 and POST 1 pass successfully, and POST 2 reports success of the module CPU tests, the PCIe card's hardware is guaranteed to be functional for basic services. In addition to POSTs, both Miniboot 0 and Miniboot 1 perform detailed, targeted tests of card hardware—cryptographic, code integrity, other infrastructure—before relying on their services.

- Segment 2: Special-purpose Linux operating system.
- Segment 3: EP11, application code, including user space drivers.

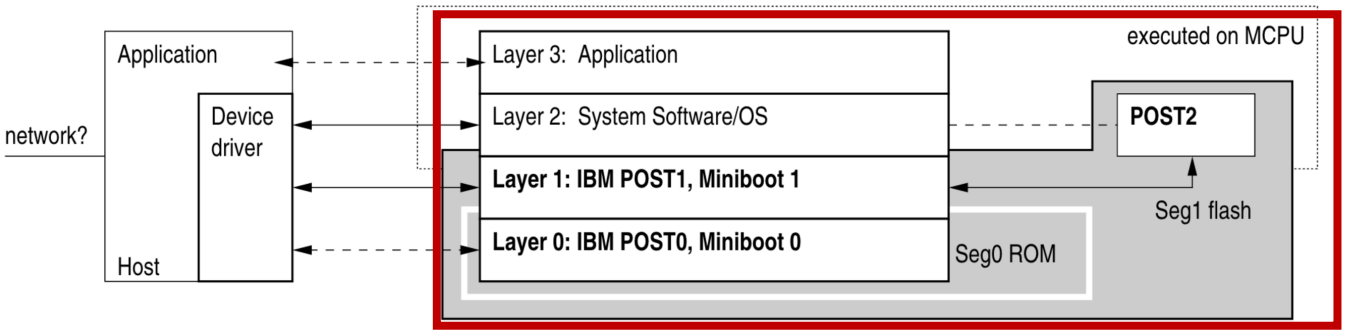

Figure 3 - Module Architecture

The module has been tested on the operational environments listed in Table 3.

Operating System	Hardware P/N	Processors
Linux based on kernel v 3.1	PN 02WN652-N37880 POST0 v9662 MB0 v6096 (Low Power)	Capri ASIC Low Power with Power PC 405 x1 Power PC 470 x2
	PN 03FM953-H07053 POST0 v8657 MB0 v6381 (Low Power)	
	PN 03JJ165-N38177 POST0 v8657 MB0 v6381 (Low Power)	
	PN 02WN654-N37880 POST0 v9662 MB0 v6096 (Standard Power)	Capri ASIC Standard Power with Power PC 405 x1 Power PC 470 x2
	PN 03JJ168-N38177 POST0 v8657 MB0 v6381 (Standard Power)	
	PN 03FM956-H07053 POST0 v8657 MB0 v6381 (Standard Power)	

Table 3: Tested Operational Environments

1.2 Modes of Operation

The module supports two modes of operation:

- **FIPS mode** (the Approved mode of operation): only approved services listed in Table 7 and Table 8 can be used.
- **Non-FIPS mode** (the non-Approved mode of operation): all approved services listed in Table 7 and Table 8 as well as non-approved services listed in Table 9 can be used in this mode. In order to switch from FIPS mode to non-FIPS mode and vice-versa, reinitialization of the module is required that zeroizes or and/or re-creates the CSPs stored in volatile and

non-volatile storage of the module such that there is no sharing of keys or CSPs between the two modes.

The Segment 3 of the module can consist of multiple domains, also called EP11 domains. The domains are the partitions that maintain their own administrative settings, key material as well as their own operational mode (FIPS or non-FIPS). The mode is reported via compliance settings using control points setup. The control points are administratively controlled sets of restrictions that enable or disable specific functionalities. When the compliance settings show that XCP_ADMS_FIPS2021 and XCP_ADMS_ADM_FIPS2021 are enabled it is identified that a domain is in FIPS mode. This mode prohibits use of any non-approved algorithm, mode key or key size. The key objects contain the expected mode of operation. Therefore, objects containing a non-expected mode of operation are not usable within the running mode.

To enable FIPS mode, the administrative setting XCP_ADMM_STR_112BIT (or a higher key strength mode i.e., XCP_ADMM_STR_128) must be enabled, and the following control points must be disabled:

XCP_CPB_KEYSZ_BELOW80BIT
XCP_CPB_KEYSZ_80BIT
XCP_CPB_ALG_RAW_RSA
XCP_CPB_SKIP_KEYTESTS
XCP_CPB_ALG_NFIPS2011
XCP_CPB_KEYSZ_HMAC_ANY
XCP_CPB_KEYSZ_RSA65536
XCP_CPB_ALG_NFIPS2021
XCP_CPB_ECDSA_OTHER
XCP_CPB_ALG_EC_25519
XCP_CPB_ALG_PQC
XCP_CPB_BTC
XCP_CPB_ALLOW_NONSESSION
XCP_CPB_ALG_EC_SECGCRV
XCP_CPB_ALG_EC_BPOOLCRV
XCP_CPB_COMPAT_LEGACY_SHA3
XCP_CPB_DSA_PARAMETER_GEN
XCP_CPB_WRAP_ASYMM
XCP_CPB_UNWRAP_ASYMM

2 Cryptographic Module Ports and Interfaces

Table 4 describes all the cryptographic module's ports and interfaces. The physical ports listed in the table below maps to the physical ports shown in Figure 2 – 4769-001 Block Diagram. All input data of the cryptographic module uses data input interfaces, and all output data of the cryptographic module uses data output interfaces. Data output is inhibited during power-on self-tests and error state.

Physical port	Logical Interface	Data that passes over port/interface
PCIe data/addresses	Data input Data output	PCIe Express signals
PCIe control	Control input Status output	PCIe Express signals
Serial ports (RS232)	Status output	Auxiliary signals
USB port	N/A (the current firmware does not use USB port)	The USB port is not used by the module.
PCIe power	Power	Auxiliary signals
Battery power (USB)	Power	Auxiliary signals
External warning (Sensor connected to the Tamper Controller)	Control input (from sensor) Status output (to host)	Auxiliary signals
N/A	EP11 Data input/ EP11 Data output	API input parameters for data/ API output parameters for data
	EP11 Control input	API function calls
	EP11 Status output	API return code

Table 4 - Ports and Interfaces

3 Roles, services, and authentication

3.1 Roles and their authentication

Table 5 describes the roles supported by the cryptographic module. The module supports the Cryptographic Officer roles and User role. The module does not support concurrent operators, and it does not support a Maintenance role.

Role	Description
CO1	Cryptographic Officer 1 Owns Segment 1 and established by IBM as the base authority.
CO2	Cryptographic Officer 2 Owns Segment 2 and established by CO1.
CO3	Cryptographic Officer 3 Owns Segment 3 and established by CO2.
CO (EP11 Module Administrator)	Performs EP11 administrative services that are opaque to the User role. Module administrators are the most privileged EP11 identities. They are authorized to submit state-changing commands at the module level.
CO (EP11 Domain Administrator)	Performs EP11 administrative services that are opaque to the User role. Domain administrators are authorized to submit state-changing commands to its own domain, but not the entire module or other domains.
User	Uses EP11 Domain services.

Table 5 - Roles Description

Table 6 lists the roles and their respective authentication methods and strengths.

Role	Authentication Method	Authentication Strength
CO1	Identity-based	ECC P-521 using SHA-512 is used for the signing and verification of digital signatures. The probability that a random attempt will succeed, or a false acceptance will occur is 1/2^256, which is less than 1/1,000,000. Even considering the rate of one (1) signature verification per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^256, which is less than 1/100,000.

CO2	Identity-based	ECC P-521 using SHA-512 is used for the signing and verification of digital signatures. The probability that a random attempt will succeed, or a false acceptance will occur is 1/2^256, which is less than 1/1,000,000. Even considering the rate of one (1) signature verification per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^256, which is less than 1/100,000.
CO3	Identity-based	ECC P-521 using SHA-512 is used for the signing and verification of digital signatures. The probability that a random attempt will succeed, or a false acceptance will occur is 1/2^256, which is less than 1/1,000,000. Even considering the rate of one (1) signature verification per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^256, which is less than 1/100,000.
CO (EP11 Module Administrator)	Identity-based	ECDSA with curves P-224, P-256, P-384, P-521 and RSA with approved keys 2048, 3072, 4096 with SHA-256 is used for the signing and verification of digital signatures. Considering lowest key size, the probability that a random attempt will succeed, or a false acceptance will occur is 1/2^112, which is less than 1/1,000,000. Even considering the rate of one (1) signature verification per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^112, which is less than 1/100,000.
CO (EP11 Domain Administrator)	Identity-based	ECDSA with curves P-224, P-256, P-384, P-521 and RSA with approved keys 2048, 3072, 4096 with SHA-256 is used for the signing and verification of digital signatures. Considering lowest key size, the probability that a random attempt will succeed, or a false acceptance will occur is 1/2^112, which is less than 1/1,000,000. Even considering the rate of one (1) signature verification per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^112, which is less than 1/100,000.
User	Identity-based	The user authentication is performed by verifying a PIN blob protected under authenticated-encryption that uses AES Key Wrapping under a 256-bit shared key derived from the SP800-56Ar3 compliance shared secret computation. The authentication strength is 2^256 , which is the security strength of the AES Key Wrapping with a 256-bit key. Even considering the rate of one (1) AES Key Unwrapping per 1µs, the probability of successfully authenticating to the Module within one minute through random attempts is 60,000,000/2^256, which is less than 1/100,000.

Table 6 - Roles and Authentication

3.2 Services

- **G** = Generates keys
- **I** = Input keys from outside of the Module
- **O** = Output Key
- **K =** Used to Encrypt/Decrypt
- **U** = Use Key
- **W** = Write/Store Key
- **Z** = Zeroize
- **R** = Read
- **D** = Delete
- **V**= Verify Signature
- **S** = Generate Signature

Table 7 lists the authenticated services supported by the cryptographic module.

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
Miniboot					
Establish Officer (CO) 2	Register new Officer 2	ECDSA	ECDSA Key (Crypto Officer1 public key)	CO1	V
		ECDSA	ECDSA Key (Device keypair (DKP1) private key)		S
Establish Officer (CO) 3	Register new Officer 3	ECDSA	ECDSA Key (Crypto Officer2 public key)	CO2	V
		ECDSA	ECDSA Keys (Device keypair (DKP1) private key)		S
Surrender Officer (CO) 2	Clear Segment 2 and 3 parameters and persistent data, and officer 2 and officer 3 public keys	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)		S
			ECDSA Key (Crypto Officer2 public key)		V, W
		ECDSA	ECDSA Key (Crypto Officer3 public key)		W
Surrender Officer (CO) 3	Clear Segment 3 parameters and persistent data and officer 3 public key	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)	CO3	S
			ECDSA Key (Crypto Officer3 public key)		V, W

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
Ordinary Burn 1	Load Segment 1 firmware and officer 1 public key; optionally clear Segment 2 and/or 3 parameters and persistent data and officer public key, as defined by Segment 2/3 persistent object definitions	ECDSA	ECDSA Key (Crypto Officer1 public key[1])	CO1	V, I, W
			ECDSA Key (Device keypair (DKP1) public key)		G, W
			ECDSA Key (Device keypair (DKP1) private key)		G, W, S
Ordinary Burn 2	Load (replace) Segment 2 firmware; optionally clear Segment 3 parameters, persistent data, and officer public key, as defined by Segment 3 persistent object definitions	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)	CO2	S
			ECDSA Key (Crypto Officer2 public key)		V
Emergency Burn 2	Clear Segment 2 and 3 parameters and persistent data and officer 2 and officer 3 public keys; Load segment 2 firmware and officer 2 public key	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)	CO1	S
			ECDSA Key (Crypto Officer1 public key)		V
			ECDSA Key (Crypto Officer2 public key)		I, W, V
Ordinary Burn 3	Load (replace) segment 3 firmware	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)	CO3	S
			ECDSA Key (Crypto Officer3 public key)		V
Emergency Burn 3	Clear Segment 3 parameters and persistent data	ECDSA	ECDSA Keys (Device keypair (DKP1) private key)	CO2	S

[1] For Ordinary Burn 1 there are two instances of the Officer1 public key – one that is already present in the Module and one that is supplied in the command. The “old” key is used to verify a signature on the command, at which point the “new” key is imported and written (replacing the “old” key).

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
	and officer 3 public key; Load Segment 3 firmware and officer 3 public key		ECDSA Key (Officer2 public key)		V
			ECDSA Key (Officer3 public key)		I, W, V
Software-Induced Tamper	Render a PCIe HSM card (i.e. the module) inoperable by evoking the module's tamper response mechanism. Evocation of this service destroys all CSPs residing on the PCIe HSM card. Note: this command is not expected to be used during the lifetime of a typical deployment since it decommissions the module and renders it useless.	ECDSA	ECDSA Key (Device keypair (DKP1) private key), DRBG seed, DRBG state	CO1	Z
			ECDSA (Officer1 public key)		V
EP11 Domain					
EC Diffie-Hellman Shared Secret Computation	Shared secret computation with EC Diffie Hellman	KAS-ECC-SSC	EC Key Pair,	EP11 User	R
			Shared Secret		W
Diffie-Hellman Shared Secret Computation	Shared secret computation with Diffie Hellman	KAS-FCC-SSC	DH Key Pair		R
			Shared Secret		W
Key Wrapping/Unwrapping	Key wrapping	AES-KW/KWP	AES Key wrapping key, wrapped key		R
Symmetric Encryption/Decryption	Symmetric encryption/decryption	AES, Triple-DES	AES, Triple-DES Keys		R
Key Generation	Key generation	DSA, ECDSA, RSA	DSA, ECDSA, RSA Keys		W
Key Verification	Key verification	ECDSA	ECDSA public key		R

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
Signature Generation/Verification	Signature generation/verification	DSA, ECDSA, RSA	DSA, ECDSA, RSA Keys		R
Random Number Generation	Random number generation	DRBG	Entropy Input		R
			Seed, Internal State		W
Message Digest	Message digest	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	N/A		N/A
Message Authentication Code (MAC)	Message Authentication Code (MAC)	AES-CMAC, Triple-DES-CMAC	AES, Triple-DES Keys		R
		HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, HMAC-SHA-512/224, HMAC-SHA-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512	HMAC Keys		R
Manage module administrator (public keys)	Replace, add or remove administrator public keys	RSA, ECDSA	RSA, ECDSA Keys, (EP11 Module Administrator Keys)	CO (EP11 Module Administrator)	V, R, W, Z, D
Export module state	Export module state by encrypting its sensitive part (wrapping keys) using AES KW	AES, RSA, ECDSA	RSA, ECDSA Keys (EP11 Module Administrator Keys)		V, R
			AES Key (EP11 Domain Wrapping Keys)		R
Import module state	Import module state that contains sensitive (wrapping keys) and non-sensitive data. Sensitive data is encrypted	AES, RSA, ECDSA	RSA, ECDSA Keys (EP11 Module Administrator Keys)		V, R
			AES Keys (EP11 Domain Wrapping Keys)		W
Zeroize Module seg 3	Zeroize all objects in segment 3 including	N/A	RSA, ECDSA Keys (EP11 Domain Administrator Keys)		D, Z

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
	anything within domains		AES Keys (EP11 Domain Wrapping Keys)		D, Z
			RSA, ECDSA Keys (EP11 Module Administrator Keys)		V, R
Manage domain administrator (public keys)	Import administrative keys with their certificates	RSA, ECDSA	RSA, ECDSA Keys (EP11 Domain Administrator Keys)	CO (EP11 Domain Administrator)	R, W
Generate importer key	Generate the importer key for importing sensitive data		RSA, ECDSA Keys (Importer Keys)		R, W
			RSA, ECDSA Keys (EP11 Module Administrator Keys)		V, R
Set domain attributes	Set the domain attributes		RSA, ECDSA Keys (EP11 Domain Administrator Keys)		V, R
Manage (set, add, remove) control points	Manage the control points to adjust the functionality of the module		RSA, ECDSA Keys (EP11 Domain Administrator Keys)		V, R
Export Wrapping Key	Export a wrapping key in encrypted form using key transport with AES KW	AES, RSA, ECDSA	AES Key (EP11 Domain Wrapping Keys)		R
			RSA, ECDSA Key (EP11 Domain Administrator Key)		V
Import Wrapping Key	Generate or import a key by generating an importer key and providing it for encryption of the wrapping key that is to be imported	AES, RSA ECDSA	AES Key (EP11 Domain Wrapping Keys)		W
			RSA, ECDSA Key (EP11 Domain Administrator Key)		V
Generate Wrapping Key		RSA ECDSA, DSA	AES Key (EP11 Domain Wrapping Keys)		W
			RSA, ECDSA Keys (EP11 Domain Administrator Keys)		V, D, Z

Service	Description	Approved Security Functions	CSPs/ Keys	Roles	Access rights to CSPs/ Keys
Zeroize Domain	Zeroizes the domain	N/A	AES Keys (EP11 Domain Wrapping Keys)		D, Z
			RSA, ECDSA Keys (EP11 Domain Administrator Keys)		D, Z

Table 7 - Authenticated Services

Table 8 lists the unauthenticated services supported by the cryptographic module.

Service	Description
Cold Boot	Reboots the Module and performs power-on self-tests
Query Status	Read infrastructure status, including segment owners. Reset the Module CPU (MCPU) (OS/application).
Query Status/Noreset	Read module status, including segment owners. Do not reset Module CPU.
Query Signed Health (“Get Health”)	Read module status, including owner identities and officer public keys; Reset Module CPU conditionally (only if Segment 2 or Segment 3 has been updated since the MCPU was last reset [in practice this is only possible for Segment 3])
Query Signed Health/No reset (“Query Firmware”)	Read module status, including owner identities and officer public keys. Do not reset Module CPU.
Query Certificate	Returns the entire Segment 1 certificate list, one certificate at a time (repeated calls to Miniboot 1).
Query Segment 0 Hash	Returns the computed SHA-512 hash of Segment 0 (Miniboot 0 concatenated with POST0).
Algorithm Test (SHA-256 test)	Compute SHA-256 hash of host-supplied data as an interactive communications/infrastructure self-test; Does not access CSPs
Continue to Segment 1	Advance from Segment 0 into Segment 1 if status permits
Continue to Segment 2	Start Segment 2 firmware if status permits
PKCS#11 Queries	Includes environment and key queries
Non-Administrative Extended Queries	Queries unique to EP11, beyond PKCS#11. General information about the module
Administrative Queries	

Service	Description
Self-Test	On-demand self-test
Show Status	Show the status of the module
Login	The service used to login to the module that performs User role authentication
Logout	The service used to logout from the module

Table 8 - Unauthenticated Services

Table 9 lists all the non-approved services used by the module.

Service	Algorithm	Key type	Role
EP11			
Key Establishment	ECDH	Brainpool/Montgomery curves	EP11 User
Key Generation	RSA	1024 bits	
	DSA	L=1024, N=160	
	ECDSA	BP192r1/t1, BP224r1/t1, BP256r1/t1, BP320r1/t1, BP384r1/t1, BP512r1/t1, secp256k1, Edwards/Montgomery curves	
Domain Parameter Generation	DSA	L=1024, N=160; L=2048, N=256; L=3072, N=256 **NOTE:** DSA Domain Parameter Generation for approved key sizes has not been ACVP-tested therefore listed as non-approved.	
Key Derivation	BIP32	secp256k1	
	SLIP10	secp256k1, ed25519, nist256	
Signature Generation/Verification	DSA	L=1024, N=160	
	EdDSA	ED25591, ED448	
	ECDSA	Brainpool, secp256k1 curves	
Signature Generation/Verification	Dilithium	1312 to 2592 bytes	

Table 9: Non-Approved Services

Table 10 lists all security functions of the module, including specific key strengths employed for approved services, and implemented modes of operation.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Miniboot				
C1187 Low Power C1188 Standard Power	AES [FIPS 197, SP 800-38A]	ECB, CBC **NOTE:** This is tested, but not used.	128, 192, 256 bits	Encryption/decryption
	AES-CMAC [FIPS 197, SP 800-38B]	CMAC **NOTE:** This is tested, but not used.	128, 192, 256 bits	MAC generation/verification
N/A	CKG [SP800-133 Rev 2]	Vendor Affirmed	ECDSA: P-521	Key generation
C1249 Low Power C1250 Standard Power	DRBG [NIST SP800-90A Rev 1]	SHA-512	N/A	Random number generation
C1249 Low Power C1250 Standard Power	ECDSA [FIPS 186-4]	SHA-512	P-521	Key generation, signature generation component /Signature verification
C1187 Low Power C1188 Standard Power	HMAC [FIPS 198-1]	SHA-256, SHA-512 **NOTE:** These SHA sizes are tested but not used: SHA-1, SHA-224, SHA-384, SHA3-224, SHA3-256, SHA3-384, SHA3-512	112 bits or greater	Message Authentication Code
C1187 Low Power C1188 Standard Power	SHS [FIPS 180-4] [FIPS 202]	SHA-256, SHA-512 **NOTE:** These SHA sizes are tested, but not used: SHA-1, SHA-224, SHA-384, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	N/A	Message digest
C1247 Low Power C1248 Standard Power	SHS [FIPS 180-4]	SHA-512	N/A	Message digest
C1187 Low Power C1188 Standard Power	Triple-DES	ECB, CBC **NOTE:** This is tested, but not used.	168 bits	Encryption/decryption

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
	Triple-DES/CMAC	CMAC **NOTE:** This is tested, but not used.		Generation/verification
EP11 Domain				
#A2470	AES [FIPS 197]	CBC, ECB	128, 192, 256 bits	Encryption/decryption
		KW, KWP	256 bits	Key wrapping/unwrapping
	AES-CMAC [FIPS 197, SP 800-38B]	CMAC	128, 192, 256 bits	Encryption/decryption
	Triple-DES [SP 800-67]	CBC, ECB	168 bits (without parity)	Decryption
	Triple-DES/CMAC [SP 800-67]	CMAC		MAC generation/verification
	HMAC	SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	112 bits or greater	Message authentication
	SHS [FIPS 180-4]	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	N/A	Message digest
N/A	CKG [SP800-133 Rev 2]	Vendor Affirmed	DSA: L=2048, N=256; L=3072, N=256 ECDSA: P-256, P-521 RSA: 2048, 3072, 4096 bits	Key generation
#A2472	SHS [FIPS 180-4]	SHA-512	N/A	Message digest
#A2470	SHA-3 [FIPS 202]	SHA3-224, SHA3-256, SHA3-384, SHA3-512	N/A	Message digest
	ECDSA [FIPS 186-4]	SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	P-224, P-256, P-384, P-521	Signature generation component, signature generation, signature verification
	ECDSA [FIPS 186-4]	Extra random bits B.4.1	P-256, P-521	Key generation

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
	KAS-ECC-SSC [Sp800-56Ar3]	ephemralUnified	P-256, P-521	Shared Secret computation
	KTS [FIPS197] [SP800-38F] [FIPS 198-1]	AES-KW, KWP	256	Key wrapping/unwrapping
		AES-CBC and HMAC-SHA-224/ HMAC-SHA-256/ HMAC-SHA-384/ HMAC-SHA-512	AES 256 HMAC with keys equal to or greater than 112 bits	
	RSA Encrypt/Decrypt (CVL)	N/A **NOTE:** This is CAVP tested, but not used.	2048, 3072, 4096	Encryption/decryption
	RSA [FIPS 186-4]	B.3.3 Random Primes that are Probably Prime	2048, 3072, 4096 bits	Key generation
	RSA [FIPS 186-4]	PSS with SHA-224, SHA-256, SHA-384, SHA-512	2048, 3072, 4096 bits	Signature generation
		PSS with SHA-1 SHA-224, SHA-256, SHA-384, SHA-512		Signature verification
		PKCS1v1.5 with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		
#A2472	DSA [FIPS 186-4]	N/A	{L=2048, N=256}, {L=3072 N=256}	Key generation
	DSA [FIPS 186-4]	SHA-256	{L=2048, N=256}, {L=3072 N=256}	Signature generation
	DSA [FIPS 186-4]	SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	{L=2048 N=256}, {L=3072 N=256}	Signature verification
	KAS-FFC-SSC [SP 800-56Ar3]	dhEphem	MODP-2048, ffdhe2048	Shared Secret computation
#A2470	KDA [SP800-56Cr2]	OneStep SHA-224, SHA-256, SHA-384, SHA-512	N/A	Key derivation

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
	KBKDF [SP800-108]	Counter HMAC-SHA2-256	N/A	Key derivation
#A2470, #A2472	Hash DRBG [SP 800-90A]	SHA-512	N/A	Random number generation
#A2413	ECDSA [FIPS 186-4]	SHA2-512	P-521	Signature generation component, signature generation, signature verification
	ECDSA [FIPS 186-4]	Testing candidate B.4.2	P-521	Key generation
	Hash DRBG [SP 800-90A]	SHA-512	N/A	Random number generation
N/A	ENT (P) SP800-90B	N/A	N/A	Random number generation

Table 10 - Approved Algorithms

Table 11 lists the algorithms that are non-approved and not allowed in the approved mode of operation.

Algorithm	Key type	Use / Function
ECDH	Brainpool/Montgomery curves	Key Establishment
DSA	L=1024, N=160	Key Generation
RSA	1024 bits	
DSA	L=1024, N=160	Signature Generation/Verification
ECDSA	BP192r1/t1, BP224r1/t1, BP256r1/t1, BP320r1/t1, BP384r1/t1, BP512r1/t1, secp256k1, Edwards/Montgomery curves	
DSA	L=1024, N=160; L=2048, N=256; L=3072, N=256	Domain Parameter Generation **NOTE:** DSA Domain Parameter Generation for approved key sizes has not been ACVP-tested therefore listed as non-approved.
BIP32	secp256k1	Key Derivation

SLIP10	secp256k1, ed25519, nist256	
EdDSA	ED25591, ED448	Signature Generation/Verification
ECDSA	Brainpool, secp256k1 curves	
Dilithium	1312 to 2592 bytes	Signature Generation/Verification

Table 11 - Non-Approved Not Allowed in the Approved Mode of Operation

3.3 Authentication

The cryptographic module supports identity-based authentication.

Interacting Miniboot, a Crypto officer is authenticated to the module for every service request. The Crypto officer signs her service request using her ECDSA private key, which is paired with her identity-encoded public key. The requested service will be provided by the module only after the CO’s signature is verified. Crypto officers prove their identities via signatures using the keys that corresponds to their identities. Therefore, they are authenticated to the module for each service they request.

Crypto Officers (EP11 Administrators) are added during initialization. During initialization, the module accepts Crypto Officers (administrators); therefore, the first administrator’s certificates will be accepted without authentication as part of the initialization and ownership establishment. As soon as enough administrators are present, and a special request is submitted, the module leaves initialization. All the subsequent requests are authenticated. Administrator commands are authenticated through public-key cryptography, while some module’s state-changing commands require signatures of multiple administrators. An administrator’s identity is proven through the possession of a signing key that corresponds to its public key. Administrator’s public keys are supplied during administrator login using X.509 certificates.

The user’s role authenticates through a token-based authentication mechanism, where the authentication token is derived from user provided PIN and session related information.

The EP11 authenticated services listed in Table 7, requires the user to present the authentication token. Upon a successful verification on the provided authentication token, the requested service is granted. The EP11 services that do not disclose, modify, substitute keys or key pairs do not require authentication. These services are listed in Table 8.

4 Physical Security

Module physical security mechanisms are mainly automatic. Intrusions, which destroy card secrets through an internal, independent action, are host-observable as system administration events. A picture of the Module security cover is presented in Figure 1.

COs may notice tamper detection through unusual Module startup, such as a card failing to initialize. It is recommended to investigate the tamper event type reported by the Module, possibly cross-checking the tamper event with other logs.

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Hard Tamper	N/A (Automatic)	N/A
Soft Tamper	N/A (Automatic)	N/A
External Warning	Module Restart	Application Discretion
Low Battery	As frequent as possible	Replace as soon as possible

Table 12 - Physical Security Inspection Guidelines

Physical security is constantly monitored through a tamper detection/ response envelope with tamper response and zeroization circuitry. No external physical monitoring is required. Environmental failure protection (EFP) is included.

A hard tamper event is caused by very high overvoltage, temperature or its rate of change out of reasonable operational range, or physical tamper (penetration of the tamper-detection matrix). Module memory-type devices (e.g., Battery Backed RAM (BBRAM), communication FIFOs) are actively zeroized. Module secrets are immediately destroyed: High Speed Erasable BBRAM (HSEB) is actively cleared at microelectronic speeds (sub-milliseconds). The Module becomes permanently inoperative: Miniboot startup does not successfully complete without secrets in HSEB.

A soft tamper event is caused by moderate overvoltage or temperature moderately out of operational range. Reaction is instantaneous. The Module is held under reset while the soft tamper conditions persist. Secrets are not destroyed.

Hard and soft temper events’ specifics are listed in Table 13.

	Temperature or voltage measurement	Specify EFP or EFT	Specify if this condition results in shutdown or zeroisation
Hard tamper event			

Low Temperature	Shipping/Storage temperature below -38°C ±3°C	EFP	Module memory-type devices (e.g., BBRAM, communication FIFOs) are actively zeroized. Module secrets are immediately destroyed: HSEB is actively cleared at microelectronic speeds (sub-milliseconds). The Module becomes permanently inoperative: Miniboot startup does not successfully terminate without secrets in HSEB.
High Temperature	Shipping/Storage temperature above +90°C ± 2°C	EFP	
Low Voltage	Dead Battery tamper threshold < 2.4V ± 0.1V on Battery Voltage	EFP	
High Voltage	High Voltage tamper threshold > 4.2V ± 0.2V on +3.3V Power supply and battery High Voltage tamper threshold > 6.28V ± 0.01V on +5V Power supply	EFP	
Low tamper event			
Low Temperature	Crypto operating temperature below 0°C ± 2°C	EFP	Reaction is instantaneous. The Module is held under reset while the soft tamper conditions persist. Secrets are not destroyed.
High Temperature	Crypto operating temperature above 83°C ± 2°C	EFP	
Low Voltage	Under voltage soft tamper threshold 4.76V ± 0.01V on +5.0V Power supply	EFP	
High Voltage	Over voltage soft tamper threshold 5.89V ± 0.05V on +5.0V Power Supply	EFP	

Table 13 - EFP/EFT

Table 14 lists the module's intended temperature range of operation. The module is tested at the low and the high temperatures of operation to pass the hardness requirement for a level 4 module.

	Hardness tested temperature measurement
Low Temperature	-42 °C
High Temperature	93 °C

Table 14 - Hardness testing temperature ranges

5 Operational Environment

The Module is designated as a limited operational environment under the FIPS 140-2 definitions. The Module includes a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated by the CMVP against FIPS 140-2 or its successor. Any other firmware other than the one listed in Table 2, loaded into this Module is out of the scope of this validation and requires a separate FIPS 140-2 validation.

6 Key Management

Table 15 describes the usage of CSPs by the cryptographic services of the module. An approved SP 800-90A compliant DRBG is used for key generation; the entropy source is contained within the module scope. The module does not output intermediate keys.

In Table 15, HSEB and Secure Flash are non-volatile memory, and DRAM is volatile memory.

CSPs/keys Name /Type	Generation	Entry /Output	Storage	Zeroization
Miniboot				
ECDSA Key (Device keypair (DKP1) private key)	FIPS 186-4 B.4.2 compliant key pair generation using unmodified DRBG output.	No entry/no output	HSEB	On hard tamper
ECDSA Key (Device keypair (DKP1) public key)	FIPS 186-4 compliant key pair generation using unmodified DRBG output.	No entry Output to the host.	Secure Flash	N/A
Entropy input	Obtained from SP 800-90B compliant entropy source	No entry/no output	DRAM	xcDRNGUninstantiate
DRBG seed and internal state	Derived from entropy input as defined in SP800-90A	No entry/no output	DRAM	xcDRNGUninstantiate
AES Key (File System Encryption Key)	Unmodified DRBG output	No entry/no output	HSEB	On hard tamper
ECDSA Key (Crypto Officer1 public key)	N/A (not generated by the module)	Entered through a CO command. Output to the host.	Secure Flash	N/A
ECDSA Key (Crypto Officer2 public key)		Entered through a CO command. Output to the host.	Secure Flash	N/A

ECDSA Key (Crypto Officer3 public key)		Entered through a CO command. Output to the host.	Secure Flash	N/A
ECDSA Key (IBM Class Root public key)	Not generated by the module	Entered during the manufacture process Output to the host.	Secure flash	N/A
EP11				
AES Key	Unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
Triple-DES	Unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
RSA key pair	FIPS 186-4 compliant key pair generation using unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
DSA key pair	FIPS 186-4 compliant key pair generation using unmodified DRBG output.	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
Entropy input	Obtained from SP 800-90B compliant entropy source	No entry/no output	DRAM	memclr()
DRBG seed and internal state	Derived from entropy input as defined in SP800-90A	No entry/no output	DRAM	memclr()
ECDSA key pair	FIPS 186-4 B.4.1 compliant key pair generation using unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()

DH key pair	SP800-56Ar3 compliant key pair generation using unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
ECDH key pair	SP800-56Ar3 compliant key pair generation using unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
HMAC key	Unmodified DRBG output	No entry/output to the host in encrypted form	DRAM/HSEB	memclr()
Shared Secret	Generated during the Diffie-Hellman or EC Diffie-Hellman shared secret computation.	No entry/output	DRAM	memclr()
ECDSA/RSA Keys (EP11 Module Administrator (CO) Public Key)	N/A	Entered with certificates/no output	Flash/DRAM	N/A
ECDSA/RSA Keys (EP11 Domain Administrator (CO) Public Key)	N/A	Entered with certificates/no output	Flash/DRAM	N/A
AES Key (EP11 Domain Wrapping Key)	Unmodified DRBG output	Can be imported in encrypted form by domain administrators using AES encryption	HSEB/DRAM	memclr()
EC/RSA Importer Key	FIPS 186-4 compliant key pair generation using unmodified DRBG output	No entry/no output for private key May output public key	DRAM	memclr()
ECDSA Keys (Operating System Keypair)	FIPS 186-4 compliant key pair generation using unmodified DRBG output	No entry/no output for private key May output public key	HSEB/DRAM	On hard tamper

Table 15 – Keys and CSPs

6.1 Random Number Generation

The module provides an [SP800-90A]-compliant Deterministic Random Bit Generator (DRBG) for creation of symmetric keys, asymmetric keys, and random number generation. The DRBG is based on SHA-512 hash function. The module performs the DRBG health tests as defined in section 11.3 of [SP800-90A]. The module uses ring oscillators (ROs) as a noise source. The entropy source is compliant with [SP800-90B] and marked as ENT (P) on the certificate. The entropy source provides full 512-bits of entropy as the input to the Hash_DRBG which uses SHA-512.

6.2 Key Generation

The Key Generation methods implemented in the module for approved services in FIPS mode are compliant with Cryptographic Key Generation (CKG) standard [SP800-133] (vendor affirmed).

The module implements symmetric key generation service for AES, Triple-DES and HMAC and asymmetric key generation services for RSA, DSA and ECDSA which are compliant with [FIPS 186-4]. The random numbers used in asymmetric and symmetric key generation are directly obtained from the [SP800-90A] Hash_DRBG.

The public and private keys used in the EC Diffie-Hellman key agreement schemes are generated internally by the module using the ECDSA key generation method compliant with [FIPS186-4] and [SP800-56Ar3]. The Diffie-Hellman key agreement scheme is also compliant with [SP800-56Ar3] and generates keys using safe primes defined in RFC7919 and RFC3526.

6.3 Key Establishment

According to Table 10 Comparable strengths in [SP800-57], the key sizes of AES, Diffie-Hellman, and EC Diffie-Hellman provide the following security strengths in FIPS mode of operation.

- Diffie-Hellman provides 112 or 128 bits of encryption strength.
- EC Diffie-Hellman provides 128 or 256 bits of encryption strength.
- AES key wrapping with AES KW and KWP key establishment methodology provides 256 bits of encryption strength.
- AES key wrapping with CBC and HMAC provides between 128 and 256 bits of encryption strength.

The module implements SP800-56ARev3 compliant DH and ECDH shared secret computation that maps to IG D.8 scenario X1(1). The module also implements key agreement scheme consisting of shared secret computation followed by SP 800-56C KDF mapping to IG D.8 scenario X1(2). However, this key agreement operation is performed internally as part of user authentication process and is not available as a service from the module.

6.4 Key Entry/Output

The module supports cryptographic AES-256 key entry and output using split knowledge based on Shamir’s Secret Sharing algorithm. The module splits the key into at least two components, which must be used to reconstruct the original key. Knowledge of any k-1 or fewer components provide no information about the original key.

CSPs that are entered, and output are encrypted with approved algorithms such as AES-CBC with HMAC or AES KW/KWP. When wrapping keys (WKs) or their parts are transported, they are encrypted with key encryption keys using the specified approved algorithms.

The module associates entered or output cryptographic keys with entities to which the keys are assigned. The association of keys with their corresponding entities is performed through authentication which described in section 3. The specification of keys that are entered into or output from the module is included in Table 15.

6.5 Key Zeroization

The module provides two types of zeroization mechanisms: zeroization to respond administrative services and zeroization to respond tamper events. The former is called firmware-induced zeroization, and the latter is called Tamper-induced zeroization.

EP11 firmware-induced zeroization can be triggered by Crypto Officer (EP11 Module Administrator) “Zeroize module seg 3” service and the Crypto Officer (EP11 Domain Administrator) “Zeroize domain” service on the need-basis. The functions for zeroization are specified in Table 15, where memclr() function is the central key clearing function.

The module also implements Tamper-induced zeroization, which can only be triggered by the module hardware in response to tamper attempts. The EP11 firmware is not involved in the Tamper-induced zeroization mechanism. In the event of tamper, keys and CSPs in the non-volatile memory HSEB and DRAM are all zeroized. Public keys stored in flash memory will not be zeroized. All the public keys are protected from the modification and substitution by the digital signatures on the public key certificates.

6.6 Key Storage

The module stores CSPs used by the cryptographic module, which are listed in Table 15, for filesystem encryption (AES Key), and keys that are used for validity of the module’s current configuration (operating system ECDSA keys) and proof of authenticity of the module (IBM Class Root ECDSA public key, DKP1 private key). These keys are stored in module’s non-volatile memory (HSEB) as specified in Table 15. The module’s internal key storage is verified by Miniboot Error Correction Code to prevent corruption caused by accidental bit flips. The tamper subsystem monitors a set of parameters to determine if a hard tamper event has occurred. In case if a tamper event has occurred, the tamper controller erases the internal key storage by overwriting it with zeroes or random values.

Additionally, the module stores CO ECDSA/RSA public keys in secure flash along with the IBM Class Root public key.

The keys used in user services including symmetric and asymmetric keys are stored in non-persistent form. In some cases, copies of keys, not including public keys, are stored in HSEB. The storage methods for all the keys used by the module are listed in Table 15.

Additionally, the module exports user keys for storage outside its cryptographic boundary in encrypted form with authenticated encryption using the approved algorithm AES with HMAC using keys listed in section 3.2. The module does not release CSPs in non-protected form.

7 EMI/EMC

The module meet the requirements of 47 CFR FCC PART 15, Subpart B, Class B (Home use).

8 Self-Tests

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

On power on or reset, the Module performs the self-tests described in the Power-On Self-tests table below. All KATs must be completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module halts and a POST error code is generated. In addition to startup tests, the Module executes conditional data tests.

8.1 Power-On Self-Tests

8.1.1 Integrity Tests

Configuration integrity test verifies firmware flash memory component and code integrity.

Non-modifiable Security Service Processor code, POST 0 and Miniboot 0, are checked for integrity, initially through an embedded non-cryptographic checksum. In case of checksum mismatch, the code halts itself (POST 0) or is not even permitted to execute (Miniboot 0, inhibited by POST 0). This code is executed only at startup, verifying that the Miniboot 0 image is not corrupted.

Once Miniboot 0 takes control, it uses the Persistent Memory Manger (PMM) to determine which copy of POST1 should be run. The PMM maintains a truncated SHA-512 hash of the contents of each segment and verifies that the hash of the chosen copy of POST1 matches the stored hash.

When POST1 runs, it performs a full PMM initialization. The PMM checks the hash on both copies of all segments and (if possible) corrects any errors detected. Uncorrectable errors cause the module to halt.

During regular operations, the crypto ASIC covers all traffic through combinations of redundant implementations, CRCs, and parity checks, in engine-specific ways. Failures are reported as specific hardware errors.

The Firmware Integrity tests are listed and described in Table 16.

Algorithm	Test
Firmware Integrity Test	
POST0 32-bit Checksum	The POST0 firmware image incorporates a 32-bit checksum computed so that when the POST0 image is treated as an array of four-byte numbers the sum of the entries is zero. POST0 copies itself from flash to RAM and then verifies the checksum on the RAM copy.
POST1 32-bit Checksum	The POST1 firmware image incorporates a 32-bit checksum computed so that when the POST1 image is treated as an array of four-byte numbers the sum of the entries is zero. When POST1 runs, it verifies the checksum on the RAM copy of itself.

Algorithm	Test
SHA-512 (truncated)	POST1 is covered by the Persistent Memory Manager (PMM). MB0 directs the PMM to decide which copy of POST1 should be loaded and run. The PMM verifies the hash of POST1 at this time.
POST2 32-bit Checksum SHA-512 (truncated)	The POST2 firmware image incorporates a 32-bit checksum computed so that when the POST2 image is treated as an array of four-byte numbers, the sum of the entries is zero. POST2 copies itself from flash to RAM and then verifies the checksum on the RAM copy. POST2 is covered by the Persistent Memory Manager (PMM). The PMM verifies the hash of POST2 when POST1 directs the PMM to initialize itself.
MB0 32-bit Checksum	The MB0 firmware image incorporates a 32-bit checksum computed so that when the MB0 image is treated as an array of four-byte numbers, the sum of the entries is zero. POST0 verifies the checksum on the copy of MB0 in flash before transferring control to MB0. While MB0 copies itself from flash to RAM, it computes the checksum and verifies that the result is zero at the end.
MB1 SHA-512 (truncated)	MB1 is covered by the Persistent Memory Manager (PMM). The PMM verifies the hash of MB1 when POST1 directs the PMM to initialize itself.

Table 16: Integrity Tests

8.1.2 Known-Answer Self-Tests

Table 17 lists the Known-Answer Self-Tests performed by the module.

Algorithm	Test
Miniboot	
AES	Encryption, Decryption Modes: ECB, CBC Keys: 128, 192, 256
	Message Authentication Modes: CMAC Keys: 128, 192, 256
HASH DRBG	Modes: SHA-512
ECDSA	Sign, Verify Modes: SHA-512

Algorithm	Test
	Keys: P-521
HMAC	SHA1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512
SHS	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512
Triple-DES	Encryption, Decryption Modes: ECB, CBC Keys: 168
	Message Authentication Modes: CMAC Keys: 168
EP11	
AES	Encryption, Decryption Modes: ECB, CBC Keys: 128, 192, 256
	Message Authentication Modes: CMAC Keys: 128, 192, 256
DH	DH SSC Hash: SHA-256 Keys: 2048
ECDH	ECDH SSC Keys: P-256, P-521
DSA	Sign/ Verify using 2048 bit key
SP 800-56Cr2 KDF (KDA)	Modes: OneStep with HMAC-SHA-256
SP 800-108 KDF	Modes: Counter with HMAC-SHA-256
ECDSA	Sign Verify Modes: SHA-256

Algorithm	Test
	Keys: P-192, P-224, P-256, P-384, P-521
HASH DRBG	Modes: SHA-512
HMAC	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512
RSA	Modes: PKCS and PSS with SHA-256 Sign, Verify Keys: 2048
SHS	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512
Triple-DES	Encryption, Decryption Modes: ECB/CBC Keys: 168
	Message Authentication Modes: CMAC Keys: 168

Table 17 – Known-Answer Self-Tests

8.1.3 Conditional Tests

Table 18 lists all the pairwise consistency tests performed by the module.

Algorithm	Test
Miniboot	
ECDSA Key Generation	Signature Generation and Verification Hash: SHA-256
EP11	
RSA Key Generation	Signature Generation and Verification Hash: SHA-256

Algorithm	Test
DSA Key Generation	Signature Generation and Verification Hash: SHA-256
ECDSA Key Generation	Signature Generation and Verification, Hash: SHA-256
SP800-90B Health Tests: RCT and APT	

Table 18 - Conditional Tests

9 Design assurance

9.1 Delivery and Operation

The module is initialized at the factory. Internal controls guarantee that each one may be initialized only once, therefore there are no field initialization requirements, other than platform-specific ones for installation of PCIe cards. Once a module has been delivered, its configuration should be logged, to verify that it is fully operational and loaded by an approved code level. Application-specific details of this verification are available outside this policy.

1. The Module will provide six (6) distinct operator roles: Cryptographic Officer 1 role, Cryptographic Officer 2, and Cryptographic Officer 3, Crypto Officer (EP11 Domain Administrator), Crypto Officer (EP11 Module Administrator), EP11 User.
2. The Module will provide identity-based authentication.
3. When the Module has not been placed in a valid role, the operator will not have access to any cryptographic services.
4. The operator will be capable of commanding the Module to perform the power on self-tests by cycling power or resetting the Module.
5. Power on self-tests do not require any operator action.
6. Data output will be inhibited during key generation, self-tests, zeroization, and error states.
7. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.
8. Please refer to section 6 for the respective zeroization methods for each CSP. Specifically, to zeroize the persistent keys the operator will have to initiate the hard tamper on the device. However, note that this will make the module inoperable.
9. The Module does not support concurrent operators.
10. The Module does not support a maintenance interface or role.
11. The Module does not support manual key entry.
12. The Module does not have any external input/output devices used for entry/output of data.
13. The Module does not enter or output plaintext CSPs.
14. The Module does not output intermediate key values.

9.2 Crypto Officer Guidance

9.2.1 Coprocessor Physical Installation

Note that on a Microsoft Windows operating system, it is necessary to install the Common Cryptographic Architecture (CCA) support software before installing a coprocessor. Other supported operating systems do not require this, but it is recommended.

To install the coprocessor into the host computer, follow these steps:

1. Locate your computer’s instructions for installing expansion cards. Throughout this procedure, follow the safety instructions in that manual.
2. Turn OFF the computer and all attached devices.

3. Disconnect all cables, including the power cable. Refer to “Danger Notice D005” in Appendix B.
4. Choose a PCIe expansion slot able to accommodate a standard short-type adapter card.
5. If the expansion slot has an individual cover, remove any bracket-holding screw and the cover.

   Attention: Electrostatic discharge (ESD) can damage the card and its components. Wear an ESD wrist-strap while handling and installing the card, or take the following precautions:

   - Limit your movements; this helps prevent static electricity from building up around you.
   - Prevent others from touching the card or other components.
   - Before removing the card from the electrostatic discharge (ESD) barrier bag, touch the bag to an unpainted metal surface on your computer and hold it there for at least two seconds.
   - Handle the card by its edges only. Do not touch exposed circuitry and components.
6. Remove the cryptographic coprocessor from its ESD barrier bag. Do not discard the bag. It can be used again whenever the coprocessor is removed from the server.
7. Insert the coprocessor into the slot; be sure that the card is fully seated.
8. If possible, install a bracket-holding screw. Some server models have a row of screws available inside the machine for this purpose.
9. Replace the host computer’s cover.
10. Reconnect the power cable and any other cables that you disconnected.
11. Turn the computer ON. The cryptographic coprocessor runs it power-on self-test (POST).

9.2.2 Firmware Installation and Entering Operational/FIPS Mode

1. Surrender and establish seg3 HSM card ownership
2. Load firmware image with Coprocessor Load Utility (CLU) tool
3. Run *CLU Status* command
4. Run *EP11Info* to show card is being initialized
5. Use Trusted Key Entry (TKE) script to bring domain out of initialization
6. Run *EP11Info* to show card has transitioned out of initialization
7. Run *EP11Info* to show domain has transitioned out of initialization
8. Use TKE to disable control points for entering FIPS mode as mentioned in section 1.2
9. Run *EP11Info* to show domain / card is in FIPS mode.

9.3 User Guidance

9.3.1 Handling Self-Test Errors

When the cryptographic module is in error state it is inactive, and all the data output and services are inhibited. Errors occurred during the self-tests and conditional tests transition the module into an error state. To recover from the error state the cryptographic module must be reset.

Error State	Cause of Error	Error Indicator
Miniboot		
ERROR STATE	AES KATs failure	Code 8003*
	Triple-DES KATs failure	
	DRBG KATs failure	
	HMAC KATs failure	
	SHA* KATs failure	
	ECDSA KATs failure	
	ECDSA PCT failure	
	POST0 checksum failure	de00a900:000000a0
	MB0 checksum failure	8003a1b0:xxxxxxxx
	POST1 checksum failure	de00a200:00001000
	MB1 checksum failure	de00a200:02001100
EP11		
ERROR STATE	ENT RCT and APT Health-tests failure	Code 80010101
	SHA* KATs failure	Hash KAT HW/SW disagree on digest / KAT: SHA* failed / KAT: hash selftests failed / KAT: selftests failed
	HMAC-SHA-* KATs failure	KAT: HMAC/SHA* f / KAT: HMAC selftests failed KAT: selftests failed
	AES KATs failure	KAT: AES/* failed / KAT: symmetric selftests failed / KAT: selftests failed / KAT: symmetric encrypt result mismatched /
	AES CMAC KATs failure	KAT: AES*-CMAC failed / KAT: CMAC selftests failed / KAT: selftests failed /
	Triple-DES KATs failure	KAT: 3DES/* failed /

Error State	Cause of Error	Error Indicator
		KAT: symmetric selftests failed / KAT: selftests failed / KAT: symmetric encrypt result mismatched
	RSA KATs failure	KAT: RSA/PSS/sign mismatched / KAT: RSA/sign-verify failed / KAT: RSA selftests failed / KAT: selftests failed / KAT: RSA selftests failed
	ECDSA KATs failure	KAT: EC/sign+verify failed / KAT: EC/sign-verify failed / KAT: EC selftests failed / KAT: selftests failed / KAT: EC/sign KAT compare failed
	DH KATs failure	KAT: DH exchange disagrees / DH: 1+2 mismatched / KAT: DH key agreement selftest failed / KAT: selftests failed
	SP800-108 KDF KAT failure	KDF self-test failed
	SP800-56Cr2 KDF KAT failure	KDF self-test failed
	DSA KAT failure	Inject Fault
	DRBG KAT failure	KAT: HW-DRNG/1 failed KAT: SW-DRNG/1 failed
	ECDSA PCT failure	CSP: signature does not verify / could not verify signing key / PK: key/gen verify failed
	DSA PCT failure	
	RSA PCT failure	
	DH KAT	PCT-Derive: derived key vals differ / could not verify deriving key / PK: key/gen verify failed
	ECDH KAT	CSP: signature does not verify / could not verify signing key / PK: key/gen verify failed / PCT-Derive: derived key vals differ / could not verify deriving key

Table 19 - Error States

9.3.2 DSA signature service usage

In the approved mode, the module does not provide DSA domain parameter generation service. For DSA, the only approved services available in the approved mode are DSA key generation, signature generation and signature verification. During the DSA signature operation, the module performs the validation of parameter "g" as required per section 4.1 of SP 800-89. the module cannot perform similar validation on parameters "p" and "q" due to unavailability of domain_parameter_seed. The module's User requesting the DSA signature service shall confirm the assurance on the validity of "p" and "q" as required per section 4.1 or 4.2 of SP 800-89.

9.4 Supplemental IBM Security Policy and Guidance

Supplemental security policy that contains additional information regarding the cryptographic module is available here.

10 Mitigation of other attacks

The module does not implement security mechanisms to mitigate other attacks.

Glossary and Abbreviations

AES	Advanced Encryption Standard
APT	Adaptive Proportion Test
BBRAM	Battery Backed RAM
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCA	Common Cryptographic Architecture
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie Hellman
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EFP	Environmental Failure Protection
EFT	Environmental Failure Testing
FIPS	Federal Information Processing Standards Publication
HMAC	Hash Message Authentication Code
HSEB	High Speed Erasable BBRAM
KAT	Known Answer Test
KEK	Key Encrypting Key
MAC	Message Authentication Code
MB	Miniboot
MCPU	Module CPU
NIST	National Institute of Science and Technology
NDRNG	Non-Deterministic Random Number Generator
PCT	Pair-Wise Consistency Test
PCIe	PCI Express Interface
PKCS	Public-Key Cryptography Standards
POST	Power-On Self-Test
RCT	Repetitive Count Test
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

SKM	Session Key Modifier
SP	Special Publication
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
TKE	Trusted Key Entry

Appendix A. References

FIPS140-2 **FIPS PUB 140-2 - Security Requirements For Cryptographic Modules**
May 2001
https://doi.org/10.6028/NIST.FIPS.140-2

FIPS140-2_IG **Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program**
May 2021
https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips140-2/FIPS1402IG.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
August 2015
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

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

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

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

FIPS202 **SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions**
August 2015
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

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

RFC3394 **Advanced Encryption Standard (AES) Key Wrap Algorithm**
September 2002
http://www.ietf.org/rfc/rfc3394.txt

RFC5649 **Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm**
August 2009
http://www.ietf.org/rfc/rfc5649.txt

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.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 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.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 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-38G	**NIST Special Publication 800-38G - Recommendation for Block Cipher Modes of Operation: Methods for Format - Preserving Encryption** March 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf
SP800-56Ar3	**NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
SP800-56Cr2	**NIST Special Publication 800-56C – Revision 1 - Recommendation for Key Derivation through Extraction-then-Expansion** August 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf
SP800-57	**NIST Special Publication 800-57 Part 1 Revision 4 - Recommendation for Key Management Part 1: General** May 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf
SP800-67r2	**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
SP800-90Ar1	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-90B	**(Second DRAFT) NIST Special Publication 800-90B - 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

SP800-131Ar1 **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

SP800-132 **NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications**
December 2010
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf

SP800-133 **NIST Special Publication 800-133 Revision 2 - 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
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf

Appendix B. Danger Notice D005

DANGER: When working on or around the system, observe the following precautions:

Electrical voltage and current from power, telephone, and communication cables are hazardous. To avoid a shock hazard:

- If IBM supplied a power cord(s), connect power to this unit only with the IBM provided power cord. Do not use the IBM provided power cord for any other product.
- Do not open or service any power supply assembly.
- Do not connect or disconnect any cables or perform installation, maintenance, or reconfiguration of this product during an electrical storm.
- The product might be equipped with multiple power cords. To remove all hazardous voltages, disconnect all power cords.
- Connect all power cords to a properly wired and grounded electrical outlet. Ensure that the outlet supplies proper voltage and phase rotation according to the system rating plate.
- Connect any equipment that will be attached to this product to properly wired outlets.
- When possible, use one hand only to connect or disconnect signal cables.
- Never turn on any equipment when there is evidence of fire, water, or structural damage.
- Do not attempt to switch on power to the machine until all possible unsafe conditions are corrected.
- Assume that an electrical safety hazard is present. Perform all continuity, grounding, and power checks specified during the subsystem installation procedures to ensure that the machine meets safety requirements.
- Do not continue with the inspection if any unsafe conditions are present.
- Disconnect the attached power cords, telecommunications systems, networks, and modems before you open the device covers, unless instructed otherwise in the installation and configuration procedures.
- Connect and disconnect cables as described in the following procedures when installing, moving, or opening covers on this product or attached devices.

  To disconnect:

  1. Turn off everything (unless instructed otherwise).
  2. Remove the power cords from the outlets.
  3. Remove the signal cables from the connectors.
  4. Remove all cables from the devices.

  To connect:

1. Turn off everything (unless instructed otherwise).
2. Attach all cables to the devices.
3. Attach the signal cables to the connectors.
4. Attach the power cords to the outlets.
5. Turn on the devices.

- Sharp edges, corners and joints may be present in and around the system. Use care when handling equipment to avoid cuts, scrapes, and pinching. (D005)