Advanced Micro Devices, Inc. (AMD)

AMD Pensando PenTrust Security Module

FIPS 140-3 Non-Proprietary Security Policy

Disclaimer

Table of Contents

List of Tables

List of Figures

Date	Version	Description
10/13/2023	1.0	Initial Release
05/16/2024	1.1	Address CMVP Comments
08/28/2024	1.2	Address CMVP Comments

1 – General

1.1 Overview

This document defines the Security Policy for AMD Pensando PenTrust Security Module, hereafter referred to as the Module. The Module meets FIPS 140-3 overall Level 1 requirements, with security levels as described in section 1.2 below.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 – Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Module is a software-hybrid module intended for use by customers seeking to implement approved cryptography.

Module Type: Software-hybrid

Module Embodiment: SingleChip

Module Characteristics:
Module is not a Sub-chip and there are no special characteristics.

Cryptographic Boundary:

The Module is a software hybrid Module that comprises of software and hardware components. The software component of the Module can only support cryptographic algorithms by utilizing the Processor Algorithm Implementations (PAI), which are enabled by software and process

service requests from the consuming application (outside of the cryptographic boundary) via the Mailbox API. The cryptographic boundary of the Module is defined by the pentrustfw.img.

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP is the physical perimeter of the AMD Pensando DPU 08-0010-01. The Module executes from the ARM Cortex M0 CPU, a single-threaded CPU, with no underlying OS. The consuming application (outside of the cryptographic boundary) executes from GPC Hardware outside of the boundary (e.g. A72 ARM Processor).

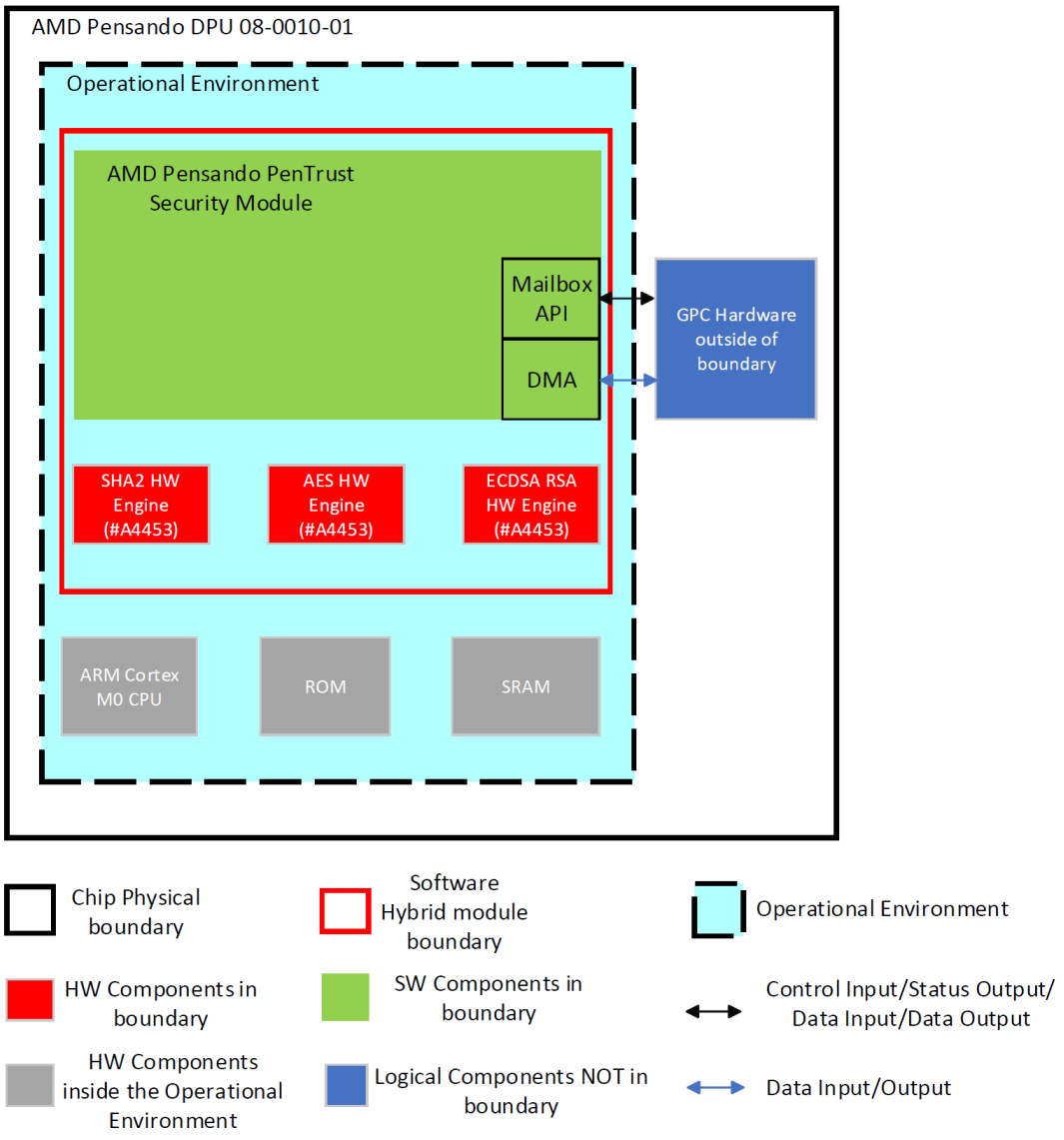

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
pentrustfw.img	5.0.0		ECDSA P-384 with SHA2-384 digital signature verification

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
AMD Pensando DPU 08-0010-01	version 01		ARM Cortex M0 CPU with PAI	

Table 3: Tested Module Identification – Hybrid Disjoint Hardware

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
N/A	AMD Pensando DPU 08-0010-01, version 01	ARM Cortex M0	Yes		5.0.0

Table 4: Tested Operational Environments - Software, Firmware, Hybrid

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

CMVP makes no statement as to the correct operation of the Module or the security strengths of the keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

The Module does not support excluded components.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode of operation	Invoke approved services of the module	Approved	Global Indicator as per FIPS 140-3 IG 2.4.C. Module only supports approved services.

Table 5: Modes List and Description

Degraded Mode Description:

The Module does not support degraded mode.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4453	Direction - Decrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4453	Direction - Decrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4453	Direction - Decrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A4453	Direction - Decrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A4453	Direction - Decrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
ECDSA SigVer (FIPS186-4)	A4453	Component - No Curve - P-384 Hash Algorithm - SHA2-384	FIPS 186-4
RSA SigVer (FIPS186-4)	A4453	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 4096	FIPS 186-4
SHA2-256	A4453	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A4453	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4453	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Digital Signature	DigSig-SigVer	Verify signatures		ECDSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) SHA2-256 SHA2-384 SHA2-512
Secure Hash	SHA	Hash Messages		SHA2-256 SHA2-384 SHA2-512
Block Cipher	BC-UnAuth	Decrypt messages		AES-CBC AES-CTR AES-ECB
Authenticated Block Cipher	BC-Auth	Decrypt and authenticate messages		AES-GCM AES-GMAC

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

There are no additional requirements for documentation of the algorithms supported by the Module.

2.8 RBG and Entropy

N/A for this module.

N/A for this module.

2.9 Key Generation

2.10 Key Establishment

2.11 Industry Protocols

The Module does not support industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Mailbox API	Data Input Data Output Control Input Status Output	All data being input or output to/from the module, control information and parameters passed via the module`s API, and status output returned from the API. All information travels through the module's API; with the module's process memory being physically in SRAM and executing on the platform's CPU. No information is transmitted by the module over a physical port on the platform.
DMA	Data Input Data Output	Data input or output resulting from Mailbox API request.

Table 8: Ports and Interfaces

3.2 Trusted Channel Specification

The Module does not support a Trusted Channel.

3.3 Control Interface Not Inhibited

The Module does not support a Control Output Interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	

Table 9: Roles

As per FIPS 140-3, the Module supports the Crypto Officer (CO) operator role implicitly. The role is implicitly assumed by the service requested. The Module does not support authentication. The Module does not support multiple concurrent operators, a maintenance role or bypass capability.

4.3 Approved Services

Please see below for the Approved Services supported by the Module. As per FIPS 140-3 IG, Section 2.4.C, a global indicator applies to this Module as it only supports Approved Services in an approved manner. An implicit indication via the successful completion of a service is the global indicator of the Module.

Please note that columns for “Input” and “Output” below are documented from the standpoint of the API parameters. It is important to note that independent of the parameters, module supports a status indicator per service to indicate success or failure. All service inputs result in a service output.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
HASH	SHA2-256 SHA2-384 SHA2-512		DataIn size, DataIn	Digest	Secure Hash	Crypto Officer
HASH_FIRST	SHA2-256 SHA2-384 SHA2-512		DataIn size, DataIn	StateOut	Secure Hash	Crypto Officer
HASH_UPD	SHA2-256 SHA2-384 SHA2-512		DataIn size, StateIn, DataIn	StateOut	Secure Hash	Crypto Officer
HASH_FINISH	SHA2-256 SHA2-384 SHA2-512		DataIn size, StateIn, DataIn	DataIn size, StateIn, DataIn	Secure Hash	Crypto Officer
VERIFY	RSA: algorithms RSA2048 /SHA2-256 and RSA4096/SHA2-512, Padding PKCS and PSS ECDSA: Curve P-384, Hash		Algorithm: RSA or ECDSA, Padding type, Hash algorithm, Key Size,	Success or Failure	Digital Signature	Crypto Officer - API RSA Key: W,E,Z - API ECDSA Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	Algorithm SHA2-384		Message Size, Public Key, Message, Signature			W,E,Z - SM Public Key: E
FIPS_GET_MODULE_VERSION	Returns the module version information. This is the show module version service required by FIPS 140-3.		DMA pointer to store result	Data containing the version information for the module		Crypto Officer
CMD_FIPS_ZEROIZE	Zeroises long lived SSPs					Crypto Officer - CM Public Key: Z - SM Public Key: Z
BOOT_SUCCESS	Used for the A72 to signal to PenTrust that boot is successful					Crypto Officer
DIAG_GET_STATUS	Reads the boot status from PenTrust. Includes which PenTrust image (0 or 1) was loaded, and which A72 image (0 or 1) was loaded			FaultStatus, Boot status info		Crypto Officer
DIAG_READPUBKEYBOOT	Used to read out the public key material for SM Public Key or CM Public Key			Key (public key of system authentication algorithm type)		Crypto Officer - CM Public Key: R - SM Public Key: R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
DIAG_SERIAL_NUMBER	Used to read out the PenTrust serial number			Serial number (128-bit)		Crypto Officer
DIAG_SET_UPGRADE_FLAG	Sets an internal PenTrust flag that will make it re-evaluate which image to boot at the next reset					Crypto Officer
REVOKE_PUB_KEY_BOOT	Used to revoke the CM Public Key or SM Public Key stored by the consuming application in GPC hardware outside the boundary		Key index (select which key is being revoked), Certificate from manufacturer, Signature of the command (based on system authentication algorithm)			Crypto Officer - CM Public Key: E - SM Public Key: E
GCM_DECR	AES 128/192/256 decrypt in GCM and GMAC modes, GMAC generation and verification		Key metadata, AAD size, Ciphertext size, Key, IV, AAD, Ciphertext, MAC	Plaintext, MAC	Authenticated Block Cipher	Crypto Officer - API AES Key: W,E,Z
DECRYPT	AES 128/192/256 decrypt in ECB, CBC, CTR		Key metadata, Ciphertext size, Key, IV or	Plaintext, Context	Block Cipher	Crypto Officer - API AES Key: W,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			context, Ciphertext			
READ_CHIP_CERT	Reads and outputs the X.509 chip certificate (binary blob) which is outside of the boundary and not an SSP. This is a helper function.			Chip Certificate		Crypto Officer
SHOW_STATUS	This is the Show Status service required by FIPS 140-3. The service is provided by the module`s API and will automatically report status of the module during Self-Tests and the Error State. See Section 10 for more information.			See Section 10 for more information		Crypto Officer
SELF_TEST	This is the Self-Tests service required by FIPS 140-3. The service is provided automatically by the module upon power-cycle, reset, or reboot. All Self-Tests are executed by the module during power-on.			See Section 10 for more information		Crypto Officer - CM Public Key: E - SW image authentication public key: E

Table 10: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The Module does not support a software load test. The Module is a hybrid software module and the loaded software image is a complete image replacement of the disjoint software component.

4.6 Bypass Actions and Status

The Module does not support Bypass Actions.

4.7 Cryptographic Output Actions and Status

The Module does not support self-initiated cryptographic output capability.

5 Software/Firmware Security

5.1 Integrity Techniques

The Module uses ECDSA P-384 with SHA2-384 Signature Verification (ECDSA Cert. #A4453) as the approved integrity technique. The Module executes an ECDSA P-384 with SHA2-384 Signature Verification Known Answer Test (KAT) prior to the software integrity test.

5.2 Initiate on Demand

To initiate the integrity test on demand the operator can power-cycle, reset, reboot the Module as per FIPS 140-3 IG 2.4.C.

5.3 Open-Source Parameters

The Module is not Open-Source.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable
The Module supports a modifiable Operational Environment.

How Requirements are Satisfied:

The modifiable Operational Environment supports a single threaded CPU where the only process that can execute at any point in time is the AMD Pensando PenTrust Security Module.

No other processes can run concurrently with the Module, and there can be only one instance of the Module. The modifiable Operational Environment does not support an operating system. Hence, the Module has control over its own SSPs, does not support uncontrolled access nor modifications to SSPs, and does not require restrictions or configurations of the operational environment for the Module to operate in an approved mode.

7 Physical Security

8 Non-Invasive Security

8.1 Mitigation Techniques

The Module does not support Non-Invasive Security. As per SP 800-140F, no additional requirements are applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
SRAM	Volatile in SRAM only	Dynamic

Table 11: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API SSP input	Entered via Mailbox API from outside the boundary	SRAM inside the boundary	Plaintext	Automated	Electronic	
PSP output	SRAM inside the boundary	Outside the boundary via Mailbox API	Plaintext	Automated	Electronic	

Table 12: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
CMD_FIPS_ZEROIZE	Service available to the consuming application to Zeroise Long lived SSPs.		API

Zeroization Method	Description	Rationale	Operator Initiation
In-line zeroise	In-line zeroisation performed automatically by the module upon completion of the service.		Automatic

Table 13: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
CM Public Key	ECDSA P-384 Public Key used to verify SW image authentication public key	384 bits - 192 bits	Public - PSP			ECDSA SigVer (FIPS186-4) (A4453)
SW image authentication public key	ECDSA P-384 Public Key used for the software integrity test.	384 bits - 192 bits	public - PSP			ECDSA SigVer (FIPS186-4) (A4453)
SM Public Key	ECDSA P-384 Public Key used to verify external A72 image.	384 bits - 192 bits	Public - PSP			ECDSA SigVer (FIPS186-4) (A4453)
API AES Key	Keys provided by the users of the API from outside the cryptographic boundary. Modes supported are: ECB, CBC, CTR, GCM, GMAC	128 bits, 192 bits, 256 bits - 128 bits, 192 bits, 256 bits	Symmetric Key - CSP			AES-CBC (A4453) AES-CTR (A4453) AES-ECB (A4453) AES-GCM (A4453) AES-GMAC (A4453)
API RSA Key	Keys provided by the users of the API from outside the cryptographic boundary for Signature Verification Services.	2048 bits, 4096 bits - 112 bits, 128 bits	Public - PSP			RSA SigVer (FIPS186-4) (A4453)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	Signature types supported are: PKCS 1.5, PKCSPSS					
API ECDSA Key	Keys provided by the users of the API from outside the cryptographic boundary for Signature Verification Services	384 bits - 192 bits	Public - PSP			ECDSA SigVer (FIPS186-4) (A4453)

Table 14: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
CM Public Key	PSP output	SRAM:Plaintext	While in use	CMD_FIPS_ZEROIZE	
SW image authentication public key		SRAM:Plaintext	While in use	N/A	
SM Public Key	PSP output	SRAM:Plaintext	While in use	CMD_FIPS_ZEROIZE	
API AES Key	API SSP input	SRAM:Plaintext	While in use	In-line zeroise	
API RSA Key	API SSP input	SRAM:Plaintext	While in use	In-line zeroise	
API ECDSA Key	API SSP input	SRAM:Plaintext	While in use	In-line zeroise	

Table 15: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
ECDSA	P-384 with SHA2-384	SW Integrity	SW/FW Integrity	Image self-test: Passed or Image self-test: FAILED	Verify

Table 16: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256	256 bits hash	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Hash	Before first use
SHA2-512	512 bits hash	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Hash	Before first use
RSA-PKCSPSS-2048	2048 bits key with SHA2-256 and Signature Type PKCSPSS	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Verify	Before first use
RSA-PKCS 1.5-2048	2048 bits key with SHA2-256 and Signature Type PKCS 1.5	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Verify	Before first use
RSA-PKCSPSS-4096	4096 bits key with SHA2-512 and Signature Type PKCSPSS	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Verify	Before first use
RSA-PKCS 1.5-4096	4096 bits key with SHA2-512 and Signature Type PKCS 1.5	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Verify	Before first use
ECDSA	P-384 with SHA2-384	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Verify	Before first use
AES-GCM	256 bit key size, GCM mode	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Decrypt	Before first use

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CTR	128 bit key size, CTR mode	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Decrypt	Before first use
AES-CBC	128 bit key size, CBC mode	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Decrypt	Before first use
AES-ECB	256 bit key size, ECB mode	KAT	CAST	KATS: algorithm_name DONE or KATS: algorithm_name FAILED	Decrypt	Before first use

Table 17: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA	SW Integrity	SW/FW Integrity	Automatically on power on	Power cycle

Table 18: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256	KAT	CAST	Automatically on power on	Power cycle
SHA2-512	KAT	CAST	Automatically on power on	Power cycle
RSA-PKCSPSS-2048	KAT	CAST	Automatically on power on	Power cycle
RSA-PKCS 1.5-2048	KAT	CAST	Automatically on power on	Power cycle
RSA-PKCSPSS-4096	KAT	CAST	Automatically on power on	Power cycle
RSA-PKCS 1.5-4096	KAT	CAST	Automatically on power on	Power cycle
ECDSA	KAT	CAST	Automatically on power on	Power cycle
AES-GCM	KAT	CAST	Automatically on power on	Power cycle
AES-CTR	KAT	CAST	Automatically on power on	Power cycle

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC	KAT	CAST	Automatically on power on	Power cycle
AES-ECB	KAT	CAST	Automatically on power on	Power cycle

Table 19: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	Module has failed a Self-Test. FIPS approved services are not provided by the module when it is in this state and data output is inhibited.	Pre-Operational Self-Tests Conditional Self-Tests	Power cycle	Image self-test: FAILED or KATS: algorithm_name FAILED

Table 20: Error States

10.5 Operator Initiation of Self-Tests

To initiate the Self-Tests on demand the operator can power-cycle, reset, reboot the Module as per FIPS 140-3 IG 2.4.C. The Module executes all Self-Tests during power-on.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Module is included inside the AMD DPU ASIC, which will be assembled together with other parts by manufacturing (AMD) into a larger product for the end user. There are no specific installation procedures or initialization procedures required for the end user.

The startup procedures of the module are:

1. Connect a console with terminal access.
2. Power on the Module.
3. Inspect the console output and confirm all Self-Tests Passed. See Section 10.1 and 10.2 above for successful indicators.
4. Issue the service FIPS_GET_MODULE_VERSION by issuing the following API:

```
./pentrust_test show_module_version
Manufacturer: AMD
Hardware name: Pensando DPU 08-0010-01
Software name: AMD Pensando PenTrust Security Module
Hardware version: 1
Hardware build: 0
Software version: 5.0.0
Crypto version: 1.0.0
```

5. Crypto Officer shall confirm the output of the Module matches the information above. Module is now ready to accept services in the Approved Mode of operation.

11.2 Administrator Guidance

The security parameters, physical ports, and logical interfaces for the Administrator (Crypto Officer) are defined via this Security Policy. Given the Module does not support authentication, the Crypto Officer role is implicitly assumed by invoking the services of the Module. The Crypto Officer is responsible for taking the following security rules into consideration:

1. The Module does not provide authentication.
2. If on-demand Self-Tests are needed, the Module must be power-cycled. All Self-Test are performed at power-up automatically.
3. Data output is inhibited during self-tests, zeroisation and error states.
4. Status information does not contain CSPs or sensitive information that if misused would lead to a compromise.
5. The Module performs both in-line zeroisation automatically for its APIs and offers the service CMD_FIPS_ZEROIZE for PSPs in SRAM.
6. The Module does not support concurrent operators.

There are no other administrative functions or security events other than what is listed above. For any questions, please contact FIPS@amd.com.

11.3 Non-Administrator Guidance

The Module does not support a Non-Administrator.

11.4 Design and Rules

Please see section 11.2 for security rules.

11.5 Maintenance Requirements

The Module does not support Maintenance.

11.6 End of Life

If the Crypto Officer would like to render the Module as no longer operable (end of life), the Crypto Officer must securely sanitize the Module by issuing the CMD_FIPS_ZEROIZE service followed by a power-cycle. Any private and public key records stored outside of the cryptographic boundary shall also be destroyed by the Crypto Officer.

12 Mitigation of Other Attacks