AMD ASP Cryptographic CoProcessor ("Raphael")

version: bc0d0346FIPS001

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.2

Last update: 2024-12-02

Prepared for:
Advanced Micro Devices (AMD)
2485 Augustine Drive
Santa Clara, CA 95054
www.amd.com

Prepared by:
atsec information security corporation
4516 Seton Center Parkway, Suite 250
Austin, TX 78759

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This section is informative to the reader to reference cryptographic services and other services of AMD ASP Cryptographic CoProcessor ("Raphael") (the “module”) from Advanced Micro Devices (AMD) (the “vendor”). Only the components listed in Section 2.1 are subject to the FIPS 140-3 validation. The CMVP (Cryptographic Module Validation Program) makes no statement as to the correct operation of the module or the security strengths of the generated keys (when supported) if the specific operational environment is not listed on the validation certificate.

The vendor has provided the non-proprietary Security Policy of the cryptographic module, which was further consolidated into this document by atsec information security together with other vendor-supplied documentation. In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

1.2 Security Levels

Table 1 describes the individual security areas of FIPS 140-3, as well as the security levels of those individual areas.

ISO/IEC 24759 Section 6 [Number Below]	FIPS 140-3 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	1
8	Non-invasive Security	Not Applicable
9	Sensitive Security Parameter Management	1
10	Self-tests	1
11	Life-cycle Assurance	1
12	Mitigation of Other Attacks	Not Applicable
Overall Level		**1**

Table 1 - Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use: The AMD ASP Cryptographic CoProcessor ("Raphael") (hereafter referred to as “the module”) supports the Ryzen PRO 7000 Series SoC (System on a Chip) by providing digital signature verification of the key database during secure boot procedures.

Module Type: Hybrid Firmware

Module Embodiment: Single-chip standalone

Module Characteristics: N/A

Cryptographic Boundary: The cryptographic boundary of the module is defined as the fips_module binary, which performs self-tests, provides the service indicator, and shows status service, as well as the hardware implementations of RSA and SHA2-384 in the Cryptographic CoProcessor (CCP), which are used to perform signature verification and verify the integrity of the fips_module binary.

Tested Operational Environment’s Physical Perimeter (TOEPP): The TOEPP of the module is defined as the Ryzen PRO 7000 Series SoC in which the module operates.

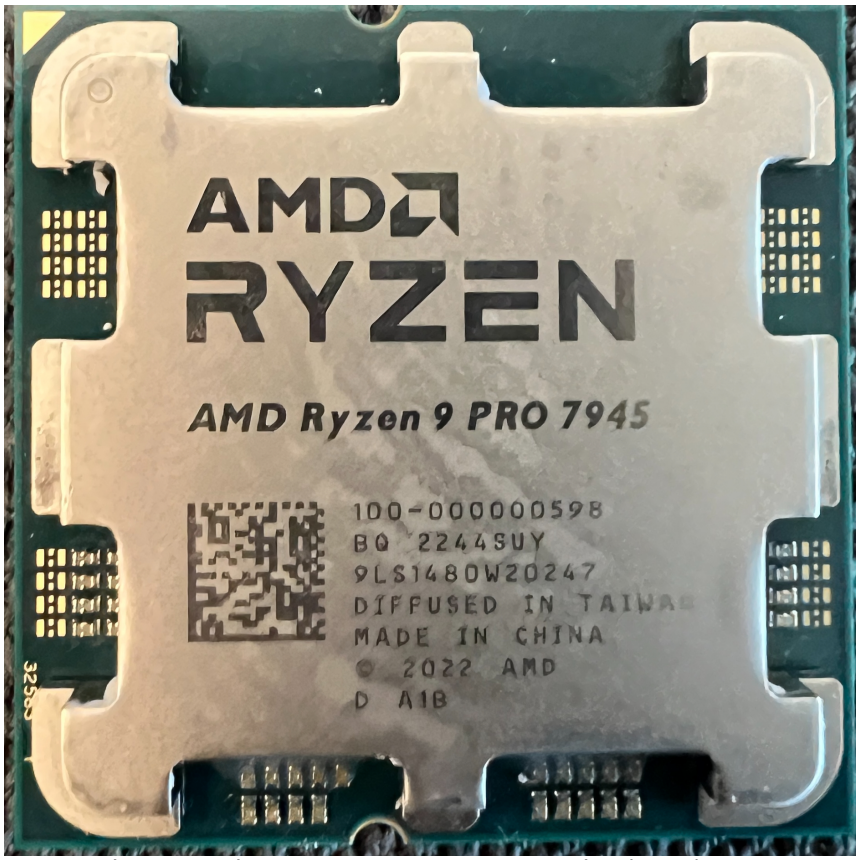

Figure 1 - The AMD Ryzen PRO 7000 Series (7945) SoC.

Figure 2 shows a block diagram that represents the design of the module. In this diagram, the physical perimeter of the operational environment, defined by the perimeter of the AMD Ryzen PRO SoC (i.e., the enclosure of the SoC), is indicated by a purple dashed line. The cryptographic boundary is represented by the components painted in orange blocks.

Components in white are only included in the diagram for informational purposes. They are not included in the cryptographic boundary (and therefore not part of the module's validation). For example, the processor is responsible for executing the non-cryptographic code in the fips_module firmware component.

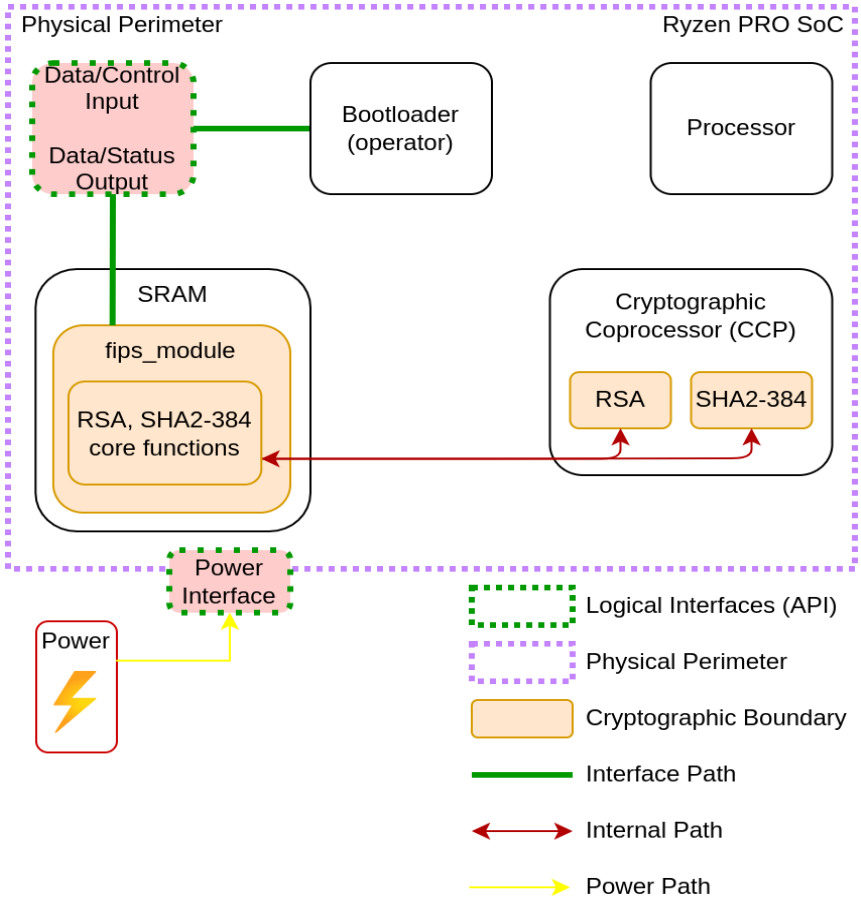

Figure 2 - Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Hardware Versions: bc0d0346FIPS001

Software Versions: N/A

Firmware Versions: bc0d0346FIPS001

Hardware Operating Environments: N/A

Software, Firmware, Hybrid Tested Operating Environments:

#	Operating System	Hardware Platform	Processor	PAA/PAI	Hypervisor and Host OS	Versions
1	N/A	AMD Ryzen PRO 7945 (100-000000598)	AMD Ryzen PRO 7945 (100-000000598)	N/A	N/A	bc0d0346FIPS001

Table 2 - Hardware Tested Operating Environments

Executable Code Sets:

Package or File Name	Software/ Firmware Version	Integrity Test Implemented
fips_module.bin	bc0d0346FIPS001	SHA2-384

Table 3 - Executable Code Sets

Vendor Affirmed Operating Environments: N/A

2.3 Excluded Components

There are no components within the cryptographic boundary that are excluded from the FIPS 140-3 security requirements.

2.4 Modes of Operation

Modes List and Description:

Name	Description	Type	Status Indicator
Approved mode	Whenever the module is operational.	Approved	The module always operates in the approved mode

Table 4 - Modes List and Description

The module implements only one mode of operation, the approved mode, in which the approved services are available. No configuration is necessary for the module to operate and remain in the approved mode.

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode.

Mode change instructions and status indicators: N/A

Degraded Mode Description: The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths	Use / Function
A4649	RSA [FIPS186-4]	PSS with SHA2-384	4096	Digital signature verification
A4649	SHA [FIPS180-4]	SHA2-384	N/A	Message digest

Table 5 - Approved Algorithms

Vendor Affirmed Algorithms: The module does not implement vendor affirmed algorithms.

Non-Approved, Allowed Algorithms: The module does not implement non-approved algorithms allowed in the approved mode of operation.

Non-Approved, Allowed Algorithms with No Security Claimed: The module does not implement non-approved algorithms allowed in the approved mode of operation with no security claimed.

Non-Approved, Not Allowed Algorithms: The module does not implement non-approved algorithms not allowed in the approved mode of operation.

2.6 Security Function Implementations

The module does not contain any approved KTS or KAS implementations.

2.7 Algorithm Specific Information

There is no algorithm specific information.

2.8 RBG and Entropy

The module does not implement any entropy sources or RBGs.

2.9 Key Generation

The module does not implement any SSP generation methods.

2.10 Key Establishment

The module does not implement any automated SSP establishment methods.

2.11 Industry Protocols

The module does not implement any industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface	Data that passes over port/interface
SRAM	Data Input	API input parameters for data.
SRAM	Data Output	API output parameters for data.
SRAM	Control Input	API function calls, API input parameters for control.
SRAM	Status Output	API return codes, status values.
Power port	Power (input) interface	Power port or pin in the single chip.

Table 6 - Ports and interfaces

Table 6 summarizes the cryptographic module interfaces. The logical interfaces are logically separated from each other by the API design. The power interface is physically separated from any other interface.

3.2 Trusted Channel Specification

The module does not implement a trusted channel.

3.3 Control Interface Not Inhibited

The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement authentication.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	N/A

Table 7 - Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module. No support is provided for multiple concurrent operators.

4.3 Approved Services

The approved service indicator can be retrieved using Microsoft HSTI and through the UEFI interactive shell tool. As the module only offers approved services, the indicator is always set when the module is operational. This is shown by the “FIPS mode: on” output.

Name	Description	Indicator	Inputs	Outputs	Security Functions	Roles	SSP Access
Digital Signature Verification	Verify a digital signature	“FIPS mode: on”	Message, public key, signature	Pass/fail	RSA PSS using SHA2-384	CO	RSA public key: W, E
Show Version	Return the module version information	None	N/A	Module version	N/A	CO	N/A
Show Status	Return the module status	None	N/A	Module status	N/A	CO	N/A
Self-test	Initiate on-demand self-tests by reset	None	N/A	Pass/fail	SHA2-384 RSA PSS	CO	N/A
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	N/A	CO	All SSPs: Z

Table 8 - Approved Services

4.4 Non-Approved Services

There are no non-approved services.

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

4.6 Bypass Actions and Status

The module does not implement a bypass capability.

4.7 Cryptographic Output Actions and Status

The module does not implement a self-initiated cryptographic output capability.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the firmware component (“fips_module.bin”) of the module is verified by comparing a SHA2-384 digest value calculated at run time with the SHA2-384 digest value stored in the module that was computed at build time.

5.2 Initiate on Demand

The integrity test is performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity tests can be invoked on demand by powering off and subsequently re-initializing the module or SoC, which will perform (among others) the firmware integrity test.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operating Environment: Non-modifiable: no changes are possible to module firmware code, nor the bootloader firmware code that interacts with the module.

How Requirements are Satisfied: The operational environment provides context separation for the memory and registers utilized by the module. When these components are used by the module, no other process or sub-component can access the information concurrently.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11.

After installation, no configuration of the operational environment is required for the module to operate in an approved mode. Therefore, there are no rules, settings, or restrictions to the configuration of the operational environment.

7 Physical Security

7.1 Mechanisms and Actions Required

The embodiment of the module is a single chip consisting of production-grade components. The coating is a standard sealing coat applied over the single chip.

The module provides no additional physical security techniques.

No actions are required to maintain the physical security of the module.

8 Non-Invasive Security

This module does not implement any non-invasive security mechanism and therefore this section is not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
SRAM	Temporary storage for SSPs used by the module as part of service execution	Dynamic

Table 9 - Storage Areas

The module does not perform persistent storage of SSPs; SSPs in use by the module exist in volatile memory only. SSPs are provided to the module by the calling process and are destroyed when released by the respective functions.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type
API input parameters	Operator residing on TOEPP	Cryptographic Module	Plaintext	Manual	Electronic

Table 10 - SSP Input-Output

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Remove power from the SoC	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed	By removing power

Table 11 - SSP Zeroization Methods

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size	Strength	Type	Generated By	Established By
RSA public key	Public key used for RSA signature verification	4096 bits	150 bits	Public key	N/A	N/A

Table 12 - SSP Information First

Name	Used By	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
RSA public key	Digital Signature Verification	API input parameters No output	RAM	While the module is operational	Remove power from the SoC	PSP	None

Table 13 - SSP Information Second

9.5 Transitions

The RSA algorithm as implemented by the module conforms to FIPS 186-4, which has been superseded by FIPS 186-5. FIPS 186-4 will be withdrawn on February 3, 2024.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm	Implementation	Test Properties	Test Method	Test Type	Indicator	Details
SHA2-384	Default	N/A	Message digest	Firmware integrity	Module becomes operational	Performed on fips_module.bin

Table 14 - Pre-Operational Self-Tests

The pre-operational firmware integrity test is performed automatically when the module is powered on before the module transitions into the operational state. While the module is executing the self-test, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after the pre-operational self-test passed successfully.

10.2 Conditional Self-Tests

Algorithm	Implementation	Test Properties	Test Method	Test Type	Indicator	Details	Condition
SHA2-384	Default	32-bit message	KAT	CAST	Module is operational	Message digest	Module initialization
RSA	Default	PSS using 4096-bit key SHA2-384	KAT	CAST		Signature verification	Module initialization

Table 15 - Conditional Self-Tests

The module performs self-tests on all approved cryptographic algorithms as part of the approved services supported in the approved mode of operation, using the tests shown in Table 15. These self-tests are performed automatically before the firmware test. Services are not available, and data output (via the data output interface) is inhibited during the self-tests. If any of these tests fails, the module transitions to the error state.

10.3 Periodic Self-Test Information

The module does not implement any periodic self-tests.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	The module immediately stops executing	SHA2-384 self-test error	Reset of the module	Error code AA0000FB
		RSA self-test error		Error code AA0000FC
		Integrity test error		Error code AA0000FD

Table 16 - Error States

In the error state, the output interface is inhibited, and the module accepts no more inputs or requests (as the module is no longer running). The error code is output through the FW status register, which explains the error that has occurred.

10.5 Operator Initiation of Self-Tests

All self-tests can be invoked on demand by unloading and subsequently re-initializing the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization and Startup Procedures

The procedures herein described are directed at OEMs for producing and configuring their BIOS so that the FIPS module is properly enabled to operate as the validated module in conformance with the rules in this Security Policy document.

Once properly installed and enabled, no configuration is necessary for the module to operate and remain in the approved mode, as it is the only mode of operation of the module.

To enable the FIPS capability

1. Reserve 16KiB at least for AMD Secure Processor level 1 directory, as the FIPS module requires additional 8KiB of ROM space for the AMD Secure Processor L1 Bootloader.
2. The Platform BIOS must include the file with “_FIPS” postfix in the file name as AMD Secure Processor entry 0x1. For example, the file PspBootLoader_stage1_prod_AB_RN_FIPS.sbin has “_FIPS” postfix in the file name. This file is thus a FIPS capable AMD Secure Processor boot loader. Conversely, the file PspBootLoader_stage1_prod_AB_RN.sbin does not have “_FIPS” postfix in the file name, making this file a non-FIPS capable AMD Secure Processor boot loader.
3. Set BIT 32 of AMD Secure Processor soft fuse chain (AMD Secure Processor entry 0xB) to enable FIPS capability.
    a. The BIT32 in AMD Secure Processor entry 0xB is defined as FIPS capability enablement. If 0, the FIPS capability is OFF; if 1, the FIPS capability is ON (i.e., the module is properly installed as the validated module described in this document).

To verify whether FIPS capability is on

1. Boot the system into UEFI shell with secure boot disabled.
2. Use the UEFI shell version of the AFF Tool version 0.3 and beyond. This tool is provided by the vendor. Run the AFF Tool with the command: afftool –fips from the interactive UEFI shell provided by the BIOS.
    a. If it shows “FIPS mode: on”, this is the FIPS capable module installed.
    b. If it shows “FIPS mode: off”, the module (described in this document) is disabled.

The screenshot in Figure 2 shows the usage of the AFF Tool. The output indicates that the FIPS module is disabled. In this condition, the module does not operate in conformance with this Security Policy document.

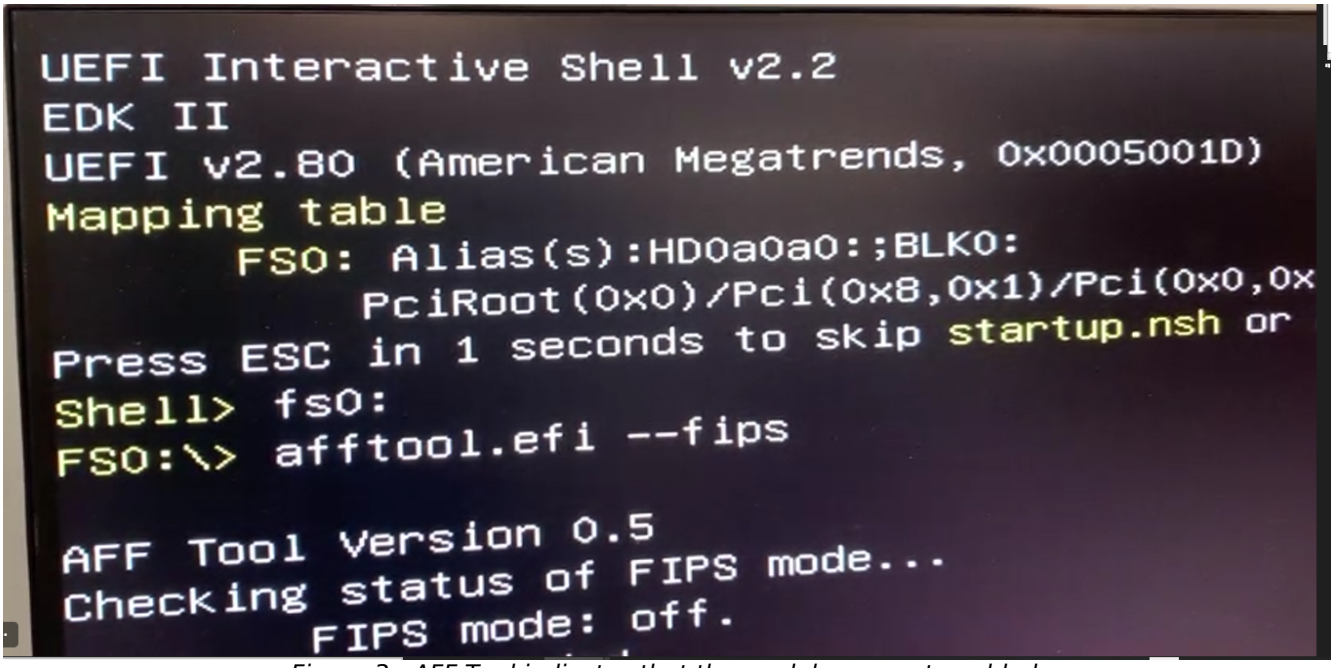

Figure 3 - AFF Tool indicates that the module was not enabled.

The screenshot in Figure 4 again shows the usage of the AFF Tool. The output demonstrates that the FIPS module is enabled and thus will operate as the FIPS validated module according to the rules in this Security Policy document.

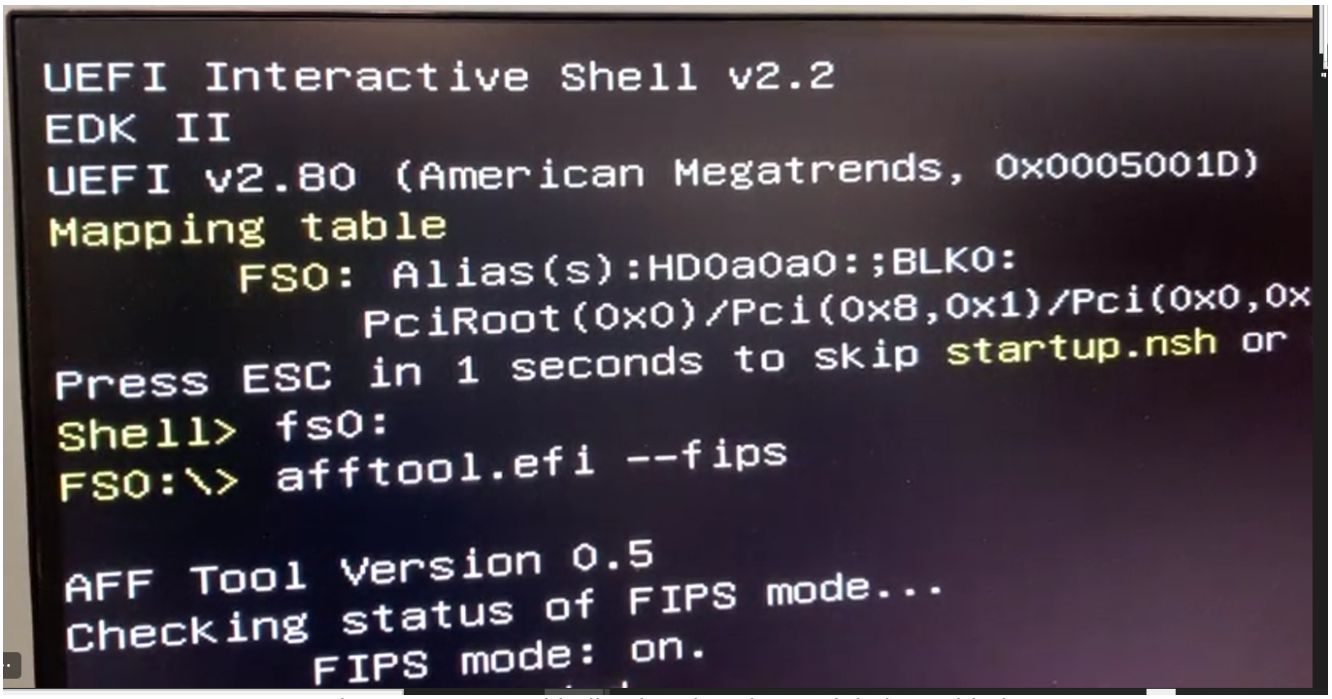

Figure 4 - AFF Tool indicating that the module is enabled.

11.2 Administrator Guidance

All the functions, ports and logical interfaces described in this document are available to the Crypto Officer. The module only provides approved functions, and as such there are no special procedures to administer the approved mode of operation.

11.3 Non-Administrator Guidance

The module implements only the Crypto Officer. There are no requirements for non-administrator operators.

11.4 Design and Rules

The bootloader (which acts as the operator of the module) initializes the fips_module component by loading it into memory upon power-on. After the pre-operational self-tests are successfully concluded, the module automatically transitions to the operational state.

In the operational state, the module automatically performs the signature verification of the key database using the RSA signature verification service, which is the sole service provided by the module. The key database, RSA public key, and signature are provided as input by the operator of the module (the bootloader). After the successful signature verification of the key database, the module unloads itself from memory, ceasing its operation.

All the procedures described above are conducted without any human assistance. To perform the procedures again, the module must be reset, which will trigger a new boot.

11.5 Maintenance Requirements

There are no maintenance requirements.

11.6 End of Life

The process for performing “End of Life” occurs at the chronological point of 10 years starting from manufacturing date of the module.

As stated in Section 9.1, the module does not possess persistent storage of SSPs. The SSP values only exists in volatile memory and those values vanish when the module is powered off. The procedure for secure sanitization of the module at the end of life is simply to power it off, which is the action of zeroization of the SSPs (Section 9.3). As a result of this sanitization via power-off, the SSPs are removed from the module, so that the module may either be distributed to other operators or disposed.

12 Mitigation of Other Attacks

The module does not offer mitigation of other attacks and therefore this section is not applicable.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
DRBG	Deterministic Random Bit Generator
FIPS	Federal Information Processing Standards
KAT	Known Answer Test
NIST	National Institute of Science and Technology
OS	Operating System
PAA	Processor Algorithm Acceleration
PSP	Public Security Parameter
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

Appendix B. References

FIPS 140-3 **FIPS PUB 140-3 - Security Requirements For Cryptographic Modules**
March 2019
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf

FIPS 140-3 IG **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**
https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements

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

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

FIPS 186-5 **Digital Signature Standard (DSS)**
February 2023
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf

SP 800-140Br1 **CMVP Security Policy Requirements**
November 2023
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf