



## **AMD ASP Cryptographic CoProcessor ("Genoa")**

**Module Version: 0000000DFIPS2001**

### **FIPS 140-3 Non-Proprietary Security Policy**

**Document Version: 1.1**

**Last update: 2025-01-08**

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

[www.atsec.com](http://www.atsec.com)

# Table of Contents

<b>0. Introduction.....</b>	<b>5</b>
0.1. Overview.....	5
0.2. This Security Policy Document.....	5
0.3. How this Security Policy was Prepared.....	5
<b>1. General.....</b>	<b>6</b>
<b>2. Cryptographic Module Specification.....</b>	<b>7</b>
2.1. Module Overview, Embodiment, Type.....	7
2.2. Module Design, Components and Versions.....	7
2.2.1. Components Excluded from Security Requirements.....	8
2.3. Security Level.....	9
2.4. Tested Operational Environments.....	9
2.5. Modes of Operation of the Module.....	9
2.6. Security Functions.....	9
2.6.1. Approved Security Functions.....	9
2.6.2. Non-Approved Security Functions Allowed in Approved Services.....	10
2.6.3. Non-Approved Security Functions Allowed in Approved Services with No Security Claimed <sup>10</sup>	
2.6.4. Non-Approved Security Functions Not Allowed in Approved Services.....	10
2.7. Rules of Operation.....	10
<b>3. Cryptographic Module Interfaces.....</b>	<b>11</b>
<b>4. Roles, Services and Authentication.....</b>	<b>12</b>
4.1. Roles.....	12
4.2. Authentication.....	12
4.3. Services.....	12
4.3.1. Approved Services.....	13
4.3.2. Non-Approved Services.....	13
<b>5. Software/Firmware Security.....</b>	<b>14</b>
5.1. Integrity Techniques.....	14
5.2. On-Demand Integrity Test.....	14
<b>6. Operational Environment.....</b>	<b>15</b>
6.1. Applicability.....	15
6.2. Tested Operational Environments.....	15
6.3. Policy and Requirements.....	15
<b>7. Physical Security.....</b>	<b>16</b>
7.1. General.....	16
<b>8. Non-Invasive Security.....</b>	<b>17</b>

<b>9. Sensitive Security Parameter Management.....</b>	<b>18</b>
9.1. SSP Generation.....	18
9.2. SSP Establishment.....	18
9.3. SSP Entry/Output.....	18
9.4. SSP Storage.....	18
9.5. SSP Zeroization.....	18
9.6. Random Number Generation.....	19
<b>10. Self Tests.....</b>	<b>20</b>
10.1. Pre-Operational Self-Tests.....	20
10.1.1. Firmware Integrity Test.....	20
10.2. Conditional Tests.....	20
10.2.1. Cryptographic Algorithm Self-Tests.....	20
10.2.2. Periodic/On-Demand Self-Test.....	20
10.3. Error States.....	21
<b>11. Life-Cycle Assurance.....</b>	<b>22</b>
11.1. Delivery and Operation.....	22
11.1.1. Secure Installation, Initialization, Start-up, and Operation.....	22
11.1.2. Maintenance Requirements.....	23
11.1.3. End of Life.....	24
11.2. Administrator Guidance.....	24
11.3. Non-Administrator Guidance.....	24
<b>12. Mitigation of Other Attacks.....</b>	<b>25</b>
<b>A. Glossary and Abbreviations.....</b>	<b>26</b>
<b>B. References.....</b>	<b>27</b>

## List of Tables

Table 1: Security Levels.....	6
Table 2: Components in the Cryptographic Boundary.....	7
Table 3: Tested Operational Environments.....	9
Table 4: Approved Algorithms.....	9
Table 5: Ports and Interfaces.....	11
Table 6: Roles, Service commands, Input, and Output.....	12
Table 7: Approved Services.....	13
Table 8: SSPs.....	18
Table 9: Self-tests.....	20
Table 10: Error States.....	21

## List of Figures

Figure 1: The AMD EPYC 9000 Series SoC, representing all versions of the single chip tested platforms.....	7
Figure 2: Block diagram.....	8
Figure 3: dmidecode indicating that FIPS module is installed.....	23
Figure 4: dmidecode indicating that FIPS module is not installed (first case).....	23
Figure 5: dmidecode indicating that FIPS module is not installed (second case).....	23

## 0. Introduction

### 0.1. Overview

This section is informative to the reader to reference cryptographic services and other services of AMD ASP Cryptographic CoProcessor ("Genoa") (the "module") from Advanced Micro Devices (AMD) (the "vendor"). Only the components listed in Section 2.2 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.

### 0.2. This Security Policy Document

This Security Policy describes the features and design of the module named AMD ASP Cryptographic CoProcessor ("Genoa")<sup>1</sup> using the terminology contained in the FIPS 140-3 specification. The FIPS 140-3 Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-3. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

The Security Policy document is one document in a FIPS 140-3 Submission Package. In addition to this document, the Submission Package contains:

- The validation report prepared by the lab.
- The Entropy Assessment Report (EAR) if applicable.
- Other supporting documentation and additional references.

This non-proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

### 0.3. How this Security Policy was Prepared

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.

---

<sup>1</sup> ASP: AMD Secure Processor

# 1. General

This document is the non-proprietary FIPS 140-3 Security Policy for version 0000000DFIPS2001 of the AMD ASP Cryptographic CoProcessor ("Genoa") cryptographic module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

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

*Table 1: Security Levels*

<b>ISO/IEC 24759 Section 6. [Number Below]</b>	<b>FIPS 140-3 Section Title</b>	<b>Security Level</b>
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
<b>Overall Level</b>		<b>1</b>

## 2. Cryptographic Module Specification

The following subsections describe the cryptographic module and how it conforms to the FIPS 140-3 specification in each of the required areas.

### 2.1. Module Overview, Embodiment, Type

The AMD ASP Cryptographic CoProcessor ("Genoa") (hereafter referred to as "the module") is defined as a hybrid firmware module in a single chip embodiment, with hardware (the coprocessor) and firmware components implementing general purpose cryptographic algorithms. The module supports the AMD EPYC 9000 Series SoC (System on a Chip) by providing digital signature verification of the key database during secure boot procedures. The module resides within the Ryzen SoC that contains the module, the processor, the firmware, and other components in a single chip embodiment (Figure 1).



*Figure 1: The AMD EPYC 9000 Series SoC, representing all versions of the single chip tested platforms.*

The Operational Environments tested for the module are described in Section 2.4

### 2.2. Module Design, Components and Versions

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 EPYC 9000 Series 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. These components are further described in Table 2.

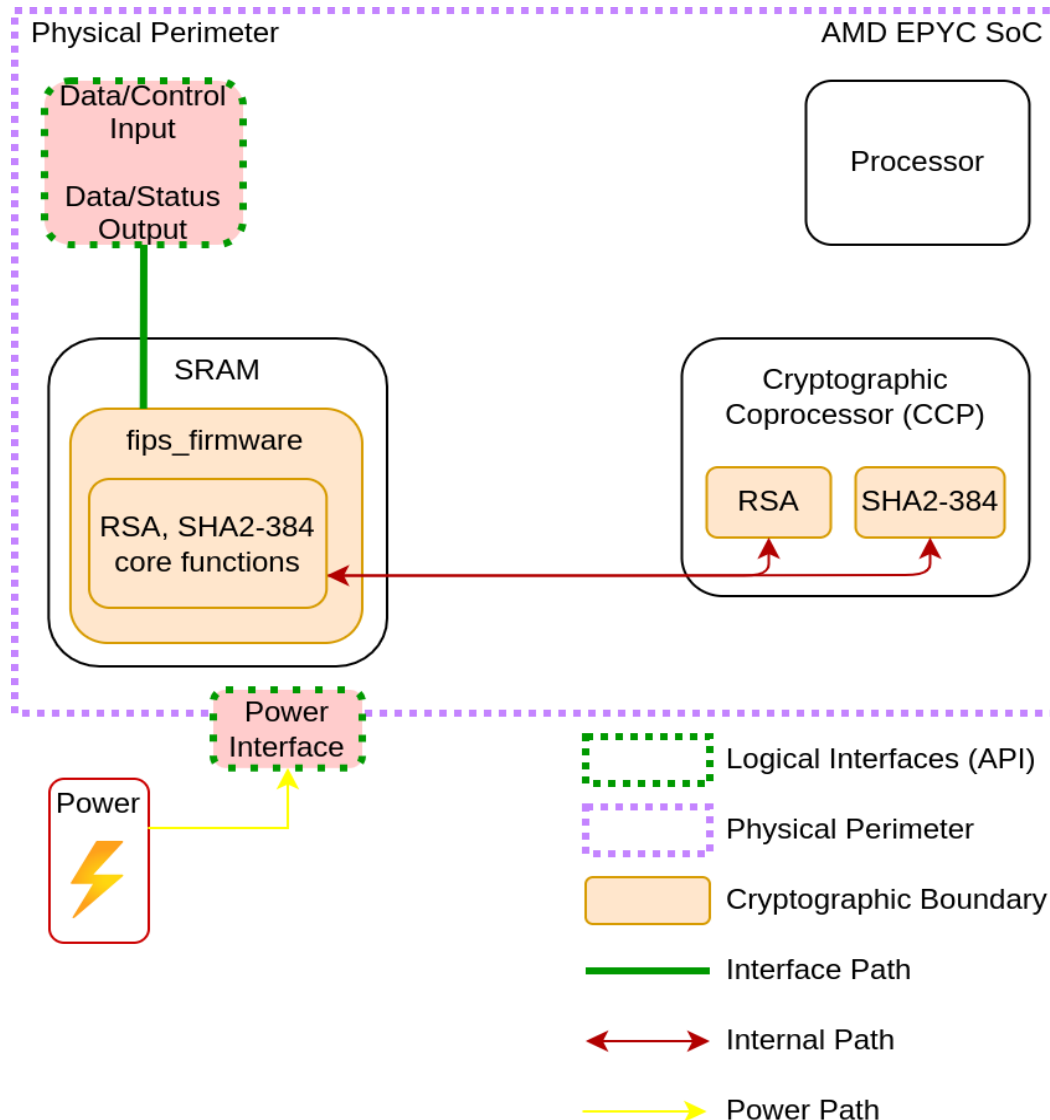
*Table 2: Components in the Cryptographic Boundary*

Component	Type	Version	Description
fips_firmware	Firmware	0000000DFIPS2001	A distinct piece of firmware which performs self-tests, provides the service indicator and show status service.
RSA implementation in the CCP	Hardware	0000000DFIPS2001	Hardware implementation of the algorithm.
SHA2-384 implementation in the CCP	Hardware	0000000DFIPS2001	Hardware implementation of the algorithm.

The flow of information between the components and the relation between that data and the module's FIPS interfaces are depicted through arrows. The arrows are colored differently to facilitate visualization. The color does not identify the type of data: the type of data flow (namely, data input, data output, status output and control input) is indicated by labels pointing to the arrows.

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_firmware` component.

Figure 2: Block diagram



### 2.2.1. Components Excluded from Security Requirements

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



## 2.3. Security Level

The module is validated according to FIPS 140-3 at overall security level 1. The security levels of individual areas are indicated in Table 1.

## 2.4. Tested Operational Environments

The module has been tested on the operational environments indicated in Table 3 with the corresponding module variants and configuration options.

*Table 3: Tested Operational Environments*

#	Operating System	Hardware Platform	Processor	PAA/ Acceleration
1	N/A	AMD EPYC 9B14 (100-000000782)	AMD EPYC 9B14 (100-000000782)	None
2	N/A	AMD EPYC 9V84 (100-000000783)	AMD EPYC 9V84 (100-000000783)	None
3	N/A	AMD EPYC 9V74 (100-000000815)	AMD EPYC 9V74 (100-000000815)	None
4	N/A	AMD EPYC 9R14 (100-000000905)	AMD EPYC 9R14 (100-000000905)	None
5	N/A	AMD EPYC 9654 (100-000000789)	AMD EPYC 9654 (100-000000789)	None
6	N/A	AMD EPYC 9554 (100-000000790)	AMD EPYC 9554 (100-000000790)	None
7	N/A	AMD EPYC 9354 (100-000000798)	AMD EPYC 9354 (100-000000798)	None

There are no vendor affirmed operational environments.

## 2.5. Modes of Operation of the Module

The module only implements one mode of operation, the approved mode, in which the approved cryptographic functions are available. The module transitions to this sole mode of operation automatically after the module completes its pre-operational self-tests and cryptographic algorithm self-tests performed on initialization. No configuration is necessary for the module to operate and remain in the approved mode.

## 2.6. Security Functions

### 2.6.1. Approved Security Functions

Table 4 lists all approved security functions (cryptographic algorithms) of the module, including specific key lengths employed for approved services, and implemented modes or methods of operation of the algorithms.

*Table 4: Approved Algorithms*

CAVP Cert.	Algorithm and Standard	Mode/Method	Description, Key Size / Key Strength	Use / Function
<a href="#">A2884</a>	RSA (FIPS 186-4)	PKCSPSS with SHA2-384	4096	Digital signature verification

CAVP Cert.	Algorithm and Standard	Mode/Method	Description, Key Size / Key Strength	Use / Function
<a href="#">A2884</a>	SHA (FIPS 180-4)	SHA2-384	N/A	Message digest

The module does not implement any vendor-affirmed algorithms.

#### **2.6.2. Non-Approved Security Functions Allowed in Approved Services**

The module does not offer any non-approved cryptographic algorithms that are allowed in approved services.

#### **2.6.3. Non-Approved Security Functions Allowed in Approved Services with No Security Claimed**

The module does not offer any non-approved cryptographic algorithms that are allowed in approved services but claim no security.

#### **2.6.4. Non-Approved Security Functions Not Allowed in Approved Services**

The module does not offer any non-approved cryptographic algorithms not allowed in approved services.

### **2.7. Rules of Operation**

The bootloader (which acts as the operator of the module) initializes the fips\_firmware component by loading it into memory upon power-on. After the pre-operational self-tests and cryptographic algorithm self-tests performed on initialization 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.

### 3. Cryptographic Module Interfaces

Table 5 summarizes the cryptographic module interfaces<sup>2</sup>. The logical interfaces are logically separated from each other by the API design. The power interface is physically separated from any other interface.

*Table 5: 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.

---

<sup>2</sup>The module does not implement a control output interface.

## 4. Roles, Services and Authentication

### 4.1. Roles

Table 6 lists the roles supported by the module with corresponding services with input and output. The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module.

*Table 6: Roles, Service commands, Input, and Output*

Role	Service	Input	Output
Crypto Officer	Digital Signature Verification	Key database (pointer to contents), signature, public key	Success, fail
Crypto Officer	Show Version	None	Name and version information in data output interface
Crypto Officer	Show Status	None	Current status in status output interface (as return codes and/or log messages)
Crypto Officer	On-Demand Self-Test	None	None
Crypto Officer	On-Demand Integrity Test	None	None
Crypto Officer	Zeroize	None	None

### 4.2. Authentication

The module does not support authentication for roles.

### 4.3. Services

The module provides services to operators that assume the available role. All services are described in detail in the user documentation.

The next subsections define the services that utilize approved, allowed, and non-approved security functions in this module. For the respective tables, the convention below applies when specifying the access permissions (types) that the service has for each SSP.

- **Generate (G):** The service establishes the SSP by generation, agreement, or derivation.
- **Read (R):** The SSP exists in the module and is read by the service, and may be output.
- **Write (W):** The caller provides the SSP to the service to be imported into the module; written; or updated if the SSP already exists in the module.
- **Execute (E) (or use):** The service uses the SSP in performing a cryptographic operation. Other access types identify the provenance of the SSP.
- **Zeroize (Z):** The service zeroizes the SSP.
- **N/A:** The service does not access any SSP or key during its operation.

The module provides only one approved service with approved parameters, and no non-approved services. The approved service indicator is thus considered a global indicator and it is set after successful completion of the pre-operational and conditional self-tests.

#### 4.3.1. Approved Services

Table 7 lists the approved services in this module, the roles that can request the service, the algorithms involved, the Sensitive Security Parameters (SSPs) involved and how they are accessed, and the respective service indicator.

In the service tables, CO specifies the Crypto Officer role.

*Table 7: Approved Services*

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Digital Signature Verification	Verify signature operations	RSA PKCSPSS using SHA2-384	RSA public key	CO	W, E	Global indicator readable through 'dmidecode' (upon successful self-tests and the module becomes operational, module only offers approved services)
<b>Other Services</b>						
Show Version	Show the version of the module's components	N/A	None	CO	N/A	None
Show Status	Show status of the module state	N/A	None	CO	N/A	None
On-Demand Self-Test	Initiate on-demand self-tests by reset	N/A	None	CO	N/A	None
On-Demand Integrity Test	Initiate the integrity test (pre-operational self-test)	SHA2-384	None	CO	N/A	Global indicator readable through 'dmidecode' (upon successful self-tests and the module becomes operational, module only offers approved services)
Zeroize	Zeroize PSP in volatile memory	N/A	All SSPs	CO	Z	None

#### 4.3.2. Non-Approved Services

There are no non-approved services.

## 5. Software/Firmware Security

### 5.1. Integrity Techniques

The integrity of the firmware component of the module ("fips\_firmware") 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. On-Demand Integrity Test

Integrity tests are performed as part of the pre-operational self-tests. The integrity test may be invoked on-demand in two ways: through the On-Demand Self-Test service, and through the On-Demand Integrity Test service.

The module provides the On-Demand Self-Test service to perform self-tests on demand. This service performs the same cryptographic algorithm tests executed during initialization, i.e., the cryptographic algorithm self-tests and the pre-operational self-tests. This service is invoked by powering-off and reloading the SoC.

The On-Demand Integrity Test service can be used to perform only the on-demand pre-operational self-tests. This service is invoked by calling the integrity test API using the module's logical interfaces. More details on the API are provided by the vendor in its developer's manual.

## **6. Operational Environment**

### **6.1. Applicability**

The module operates in a non-modifiable operational environment per FIPS 140-3 level 1 specifications: no changes are possible to module firmware code, nor the bootloader firmware code that interacts with the module.

### **6.2. Tested Operational Environments**

Please see Section 2.4.

### **6.3. Policy and Requirements**

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.

The bootloader acts as the sole operator of the module, thus there are no concurrent operators.

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.

The module does not have the capability of loading software or firmware from an external source.

## **7. Physical Security**

### **7.1. General**

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**

The module claims no non-invasive security techniques.

## 9. Sensitive Security Parameter Management

Table 8 summarizes the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module in the approved services (Table 7).

Table 8: SSPs

Key / SSP Name / Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use & related keys
RSA public key	150 bits	RSA PKCSPSS signature verification ( <a href="#">A2884</a> )	N/A	Input in plaintext through data input interface (MD/EE).  No output.	N/A	Volatile memory	When the module ends its operation	RSA signature verification

### 9.1. SSP Generation

The module does not generate SSPs.

### 9.2. SSP Establishment

The module does not implement automated SSP establishment.

### 9.3. SSP Entry/Output

The module only supports manual distribution, electronic entry of the RSA public key, which is provided in plaintext by the bootloader operator via the data input interface.

No other SSPs are entered into the module. No SSPs are output from the module.

### 9.4. SSP Storage

SSPs are provided to the module by the calling process and are destroyed when released by the respective functions.

The module does not perform persistent storage of SSPs; keys in use by the module exist in volatile memory only.

### 9.5. SSP Zeroization

The module's functions deallocates and zeroizes temporary SSP values in volatile memory used during the function's execution. The zeroization consists of writing zeroes to the memory location used by the SSP before deallocating the area. The module does not overwrite the SSP with another SSP.

The zeroization service for the SSP in volatile memory consists of powering off the SoC, which will remove power from the volatile memory. This action will cause the value of the SSP in volatile memory to be overwritten by random values. The successful act of powering off the SoC serves as the implicit indicator of zeroization.

## 9.6. Random Number Generation

The module does not implement random number generation.

## 10. Self Tests

The module performs pre-operational self-tests and conditional self-tests. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed.

All the self-tests are listed in Table 9, with the respective condition under which those tests are performed. The firmware integrity test is performed after all conditional algorithm self-tests (CASTs) are performed.

*Table 9: Self-tests*

Algorithm	Parameters	Condition for Test	Type	Test
SHA2-384	N/A	Firmware integrity test on firmware component at initialization (after all CASTs)	Pre-Operational Self-Test	Digest verification on firmware component
SHA2-384	N/A	Initialization	Conditional Algorithm Self-Test	KAT SHA2-384
RSA	SHA2-384 and 4096-bit key	Initialization	Conditional Algorithm Self-Test	KAT signature verification

### 10.1. Pre-Operational Self-Tests

The module performs pre-operational tests automatically when the module is initialized. The pre-operational self-tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

The types of pre-operational self-tests are described in the next sub-sections.

#### 10.1.1. Firmware Integrity Test

The integrity of the firmware component of the module ("fips\_firmware") 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. If the comparison verification fails, the module transitions to the error state (Section 10.3). The SHA2-384 algorithm goes through its conditional algorithm self-test before the integrity test is performed (Table 9).

### 10.2. Conditional Tests

#### 10.2.1. Cryptographic Algorithm Self-Tests

The module performs self-tests on all approved cryptographic algorithms automatically on initialization, using the tests shown in Table 9. Data output through the data output interface is inhibited during the self-tests.

#### 10.2.2. Periodic/On-Demand Self-Test

The module performs on-demand self-tests initiated by the operator, by powering off and powering the SoC back on. The full suite of self-tests in Table 9 is then executed.

The same procedure may be employed by the operator to perform periodic self-tests.

### 10.3. Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the module outputs the error type through the status indicator and status output interface. In the error state, the data output interface is inhibited, and the module accepts no more inputs or requests. The module does not implement a control output interface.

Table 10 lists the error state and the status indicator (through the FW\_STATUS variable) values that explains the error that has occurred.

*Table 10: Error States*

Error State	Error Condition	Status Indicator (FW_STATUS)
Error	SHA2-384 self-test error	Error code AA0000FB
	RSA self-test error	Error code AA0000FC
	Integrity test error	Error code AA0000FD

To recover from the error state (clearing the error condition), the module shall be restarted or reset.

## 11. Life-Cycle Assurance

### 11.1. Delivery and Operation

#### 11.1.1. Secure Installation, Initialization, Start-up, and Operation

The procedures herein described are directed at OEMs for producing and configuring their BIOS so that the FIPS module is properly installed 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 at least 16KiB for the AMD Secure Processor level 1 directory, as the FIPS module requires additional 8KiB of ROM space for the AMD Secure Processor L1 Bootloader.
2. Set BIT 32 of the 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).

Please note that above configuration is only applicable for OEMs integrating the module in as part of the production process. Once the module is distributed to end users, no further installation is required. Also, it is not possible to disable the FIPS capability once the SoC has left the OEM production process.

##### To verify whether FIPS capability is enabled

1. Boot the system into a Linux shell.
2. If the system fails to boot: one of the self-tests failed, and the module is stuck in the error state.
3. Otherwise, use the 'dmidecode' tool with type 11 as the root user to fetch FIPS status: `'sudo dmidecode -t 11'`:
  - a. If it shows "FipsStatus: PASS", the FIPS capable module is installed and has passed all required self-tests.
  - b. If it shows "FipsStatus: FAIL" or "FipsStatus:" the module (described in this document) is disabled or not installed correctly.

The screenshot in Figure 3 shows the usage of dmidecode. The output indicates that FIPS module is installed and thus will operate as the FIPS validated module according to the rules in this Security Policy document. Note that the module version is shown here as "FipsId". The "CcpVersion" string is unrelated to the FIPS requirements and should be disregarded.

Figure 3: dmidecode indicating that FIPS module is installed.

```
Handle 0x001B, DMI type 11, 5 bytes
OEM Strings
    String 1: Default string

Handle 0x0031, DMI type 11, 5 bytes
OEM Strings
    String 1: FipsStatus: PASS
    String 2: CcpVersion: 0802EF4D
    String 3: FipsId: 0000000DFIPS2001
```

The screenshots in Figure 4 and Figure 5 again shows the usage of dmidecode. The output demonstrates that FIPS module is not installed. In this condition, the module does not operate in conformance with this Security Policy document. Note that “FipsStatus” will either show “FAIL” or a blank string to indicate FIPS module is not installed. There is no functional difference between these two outputs. Finally, note that “FipsId” is also blank in this case.

Figure 4: dmidecode indicating that FIPS module is not installed (first case).

```
Handle 0x001B, DMI type 11, 5 bytes
OEM Strings
    String 1: Default string

Handle 0x0031, DMI type 11, 5 bytes
OEM Strings
    String 1: FipsStatus: FAIL
    String 2: CcpVersion: 0802EF4D
    String 3: FipsId:
```

Figure 5: dmidecode indicating that FIPS module is not installed (second case).

```
Handle 0x001B, DMI type 11, 5 bytes
OEM Strings
    String 1: Default string

Handle 0x0031, DMI type 11, 5 bytes
OEM Strings
    String 1: FipsStatus:
    String 2: CcpVersion: 0802EF4D
    String 3: FipsId:
```

### 11.1.2. Maintenance Requirements

There are no maintenance requirements.

### **11.1.3. 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 9.4, the module does not possess persistent storage of SSPs. The SSP value only exists in volatile memory and that value vanishes when the SoC 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 9.5) . As a result of this sanitization via power-off, the SSP is removed from the module, so that the module may either be distributed to other operators or disposed.

## **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.



## 12. Mitigation of Other Attacks

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

## A. Glossary and Abbreviations

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

## B. References

- Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program. (2022, March 14). *Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program*. Retrieved 03 8, 2021, from [https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips 140-3/FIPS 140-3 IG.pdf](https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf)
- National Institute of Standards Technology. (2019, March). FIPS PUB 140-3. Security Requirements for Cryptographic Modules. *FIPS PUB 140-3. Security Requirements for Cryptographic Modules*. Retrieved from <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf>
- FIPS PUB 180-4. Secure Hash Standard (SHS). (2012, March). *FIPS PUB 180-4. Secure Hash Standard (SHS)*. Gaithersburg, MD 20899-8900: National Institute of Standards & Technology. Retrieved from <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>
- FIPS PUB 186-4. Digital Signature Standard (DSS). (2013, July). *FIPS PUB 186-4. Digital Signature Standard (DSS)*. <https://doi.org/10.6028/NIST.FIPS.186-4>. doi:<https://doi.org/10.6028/NIST.FIPS.186-4>
- ISO/IEC. (2012, August). ISO/IEC 19790:2012 Information technology — Security techniques — Security requirements for cryptographic modules. *ISO/IEC 19790:2012 Information technology — Security techniques — Security requirements for cryptographic modules*. Retrieved from <https://www.iso.org/standard/52906.html>
- ISO/IEC. (2017, March). ISO/IEC 24759:2017 Information technology — Security techniques — Test requirements for cryptographic modules. *ISO/IEC 24759:2017 Information technology — Security techniques — Test requirements for cryptographic modules*. Retrieved from <https://www.iso.org/standard/72515.html>