# Pavilion Cryptographic Module FIPS 140-2 Security Policy

**Version 1.2**

**Last Updated:** July 2021
**Document Version:** Version 1.2

**Legal Notice:**



**Technical Support:**

Technical Support maintains support centre's globally. If you have questions regarding an existing support agreement, please email/phone the support agreement administration team as follows:

**Phone:**

**USA & Canada**: 1-888-342-0461

**United Kingdom**: 0800–69-8055

**Netherlands**: 0800-022-2832

**International**: 1-408-684-4958

**Email:**

support@pavilion.io

**Documentation:**
Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2.

**Documentation Feedback:**
Your feedback is important for us.
For documentation feedback send your comments to your Pavilion field support contact.

# Table of Contents



# Introduction

The Pavilion Data Systems Pavilion Cryptographic Module (S.W. Version: 1.0, H.W. Version: Intel Xeon D-1548) is a software-hybrid multi-chip standalone module designed to protect data at rest for storage volumes stored on the Pavilion Hyperparallel Flash Array. The software component of the module is a C-language application programming interface (API). The hardware component of the module is the CPU which supports the AES-NI instruction set, which is invoked for AES operations performed by the module.

# Cryptographic Boundary

The following diagram defines the cryptographic boundary:

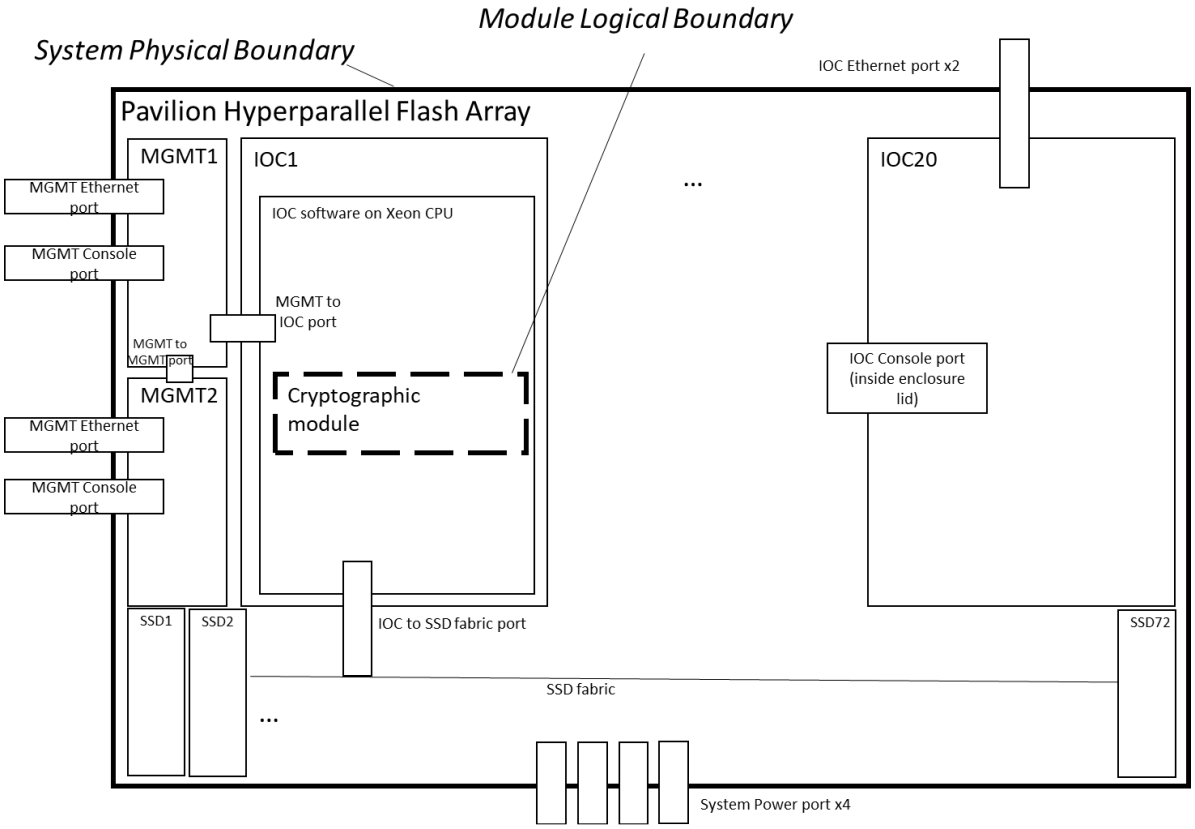


Exhibit 1 – *Specification of Cryptographic Boundary*

The physical boundary of the Pavilion Cryptographic Module is the physical boundary of the device that contains it.

# Acronyms

| TERM | DESCRIPTION |
|---|---|
| **API** | Application Programming Interface |
| **AES** | Advanced Encryption Standard |
| **CSP** | Critical Security Parameter |
| **CO** | Cryptographic Officer |
| **SHA** | Secure Hash Algorithm |
| **CMVP** | Cryptographic Module Validation Program |
| **ECDSA** | Elliptic Curve Digital Signature Algorithm |

Exhibit 2 – *Specification of Acronyms and their Descriptions*

## Security Level Specification

Fill in the security levels corresponding to each area:

| Security Requirements Area | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles, Services, and Authentication | 1 |
| Finite State Model | 1 |
| Physical Security | 1 |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 1 |
| Self-tests | 1 |
| Design Assurance | 1 |
| Mitigation of Other Attacks | N/A |

Exhibit 3 – *Security Level Table.*

## Physical Ports and Logical Interfaces

The physical ports for the Pavilion Cryptographic Module are the same as the device that it executes on. The logical interface is a C-language API.

| FIPS Interface | Logical Interface | Physical Port |
|---|---|---|
| Data Input | API Input Parameters | IOC Ethernet Port (Qty:2), IOC to SSD fabric port |
| Data Output | API Data Parameters | IOC Ethernet Port (QTY:2), IOC to SSD fabric port |
| Control Input | API Data Return Values | MGMT to MGMT port, MGMT to IOC port |
| Status Output | API Status Return Values | MGMT to MGMT port |
| Power Input | N/A | Power supply port (Qty:4) |

Exhibit 4 – *Specification of Cryptographic Module Physical Ports and Logical Interfaces*

As a software module, control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in an error state, all output on the module's logical data output interfaces is inhibited.

## Security Rules

The following specifies the security rules under which the cryptographic module shall operate:

1. The module performs the following self-tests:
    a. Power-Up Self-Tests:
        i. AES-XTS 128 Encrypt KAT
        ii. AES-XTS 128 Decrypt KAT
        iii. AES-XTS 256 Encrypt KAT
        iv. AES-XTS 256 Decrypt KAT
        v. AES CBC 128 Encrypt KAT
        vi. AES CBC 128 Decrypt KAT
        vii. AES CBC 256 Encrypt KAT
        viii. AES CBC 256 Decrypt KAT
    b. Software Integrity Test (ECDSA P-256 SHA-256 Verification)
2. The module does not provide access to CSPs until the operator is in a valid role.
3. The operator is capable of commanding the module to perform the power-up self-tests by cycling power or resetting the module.
4. Data output is inhibited during self-tests, zeroization, and error states.
5. Status information does not contain CSPs of sensitive data that if misused could lead to a compromise of the module
6. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
7. The module does not support a maintenance interface or role.
8. The module does not support manual key entry.
9. The module does not enter or output plaintext CSPs.
10. The module enforces logical separation of all data inputs, data outputs, control inputs, and status outputs.
11. The following items shall not be performed by the module user:
    a. Shall not use kernel debugger

b. Shall not output CSPs from the physical boundary of the module in plaintext via automated processes
c. Shall not store CSPs outside the logical boundary of the module in a manner that could lead to a compromise.

## CSPs, Public Keys & Private Keys

The following CSPs are included in the Pavilion Cryptographic Module:

| Description | Type | Generation | Establishment | Entry | Output | Storage | Zeroization |
|---|---|---|---|---|---|---|---|
| Volume Keys | AES-XTS 128 and 256 Bit Keys | N/A (Outside of the module via KMIP, user entry, or internal generation. Out-of-scope) | N/A | N/A - Passed in plaintext from consuming application (IG 7.7) | N/A | Plaintext in DRAM | Reboot, pds_crypt_deregister_key() API, module shutdown. |
| Firmware Integrity ECDSA Public Key | ECDSA P-256 SHA-256 | N /A (outside of the module on build server) | N/A | N/A - Public key embedded in the compiled module. | N/A | Plaintext and baked into the SW binary inside of Ferri SATA SSD IC | N/A |

Exhibit 5 – *Specification of Critical Security Parameters*

# Identification and Authentication Policy

The module supports the User and Cryptographic Officer (CO) roles. The module does not support authentication mechanisms. The roles have identical functionality. The Cryptographic Officer is the human officer, physically present, responsible for Module initialization and installation.

Only one role can be active at a time and the module does not allow concurrent operators. The module does not support a maintenance role.

| Role | Authentication Type | Authentication Data |
|---|---|---|
| Cryptographic Officer | Not Required | Not Required |
| User | Not Required | Not Required |

Exhibit 6 - *Roles and Required Identification and Authentication (FIPS 140-2 Table C1)*

The Cryptographic Officer shall perform the installation and initialization of the module as follows:

The Module is a software module that comes pre-installed in each IO Controller delivered from Pavilion Data Systems, either within a new chassis or as a stand-alone delivery (e.g. chassis expansion or parts replacement).

If the system software needs to be re-installed or upgraded, the Module software will be bundled with it, all delivered in the form of a single file with a name of the form:

*PDOS_2.7.0.0_B2_R11562-R9507-R9416-09-01-20-03-26-55.iso*

This single file should be installed by following the regular Pavilion upgrade procedure, which involves:

- Entering the chassis GUI
- Navigating to the ‘Configuration’ tab
- Navigating to the ‘Upgrade’ item
- Clicking the ‘Upgrade’ button
- Selecting the required node type to upgrade

- Selecting the given .iso file from the local system as the package to upgrade
- Clicking the ‘Next’ button, and following the status for automated installation.

Alternatively, the equivalent ‘upgrade’ CLI command can also be used.

The Module will always start in the secure Approved Mode, there is no additional action required to enter that secure mode. As part of this, the module will always run its power on self-tests to verify secure operation.

The System containing the module ships with the Cryptographic Officer’s username set to ‘admin’ and the password set to ‘admin’. This password should be changed at initialization time. This is done by logging in to the system GUI with a web browser as the ‘admin’ user, clicking on the ‘admin’ user in the top-right corner, selecting ‘Change password’ and following the prompts. The username and password are implemented by the operating system and are not within the logical boundary of the module.

The version of the Module in use can be verified by observing the following log message in the IO controller’s console log:

*pds_crypt module v1.0 initializing*

The current version is v1.0.

Successful completion of these self-tests indicate that the Module is in a secure operational state. This can be verified by either:

1. Observing that the IO controller does not reboot, or
2. Observing the following log message in the IO controller’s console log:

*pds_crypt module initialization completed successfully*

If any power-on self-test were to fail, the module will cause the affected IO controller to log a message and reboot. This can be verified by either:

1. Observing that the IO controller does reboot, or
2. Observing the following log message in the IO controller’s console log:

*pds_crypt KAT selftest failed*

If the module's integrity check were to fail, the module will cause the affected IO controller to log a message and reboot. This can be verified by either:

1. Observing that the IO controller does reboot, or
2. Observing the following log message in the IO controller's console log:

*pds_crypt integrity check failed*

The kernel will panic upon the module entering the error state, and the console log will display "kernel panic".

## Access Control Policy

| Service | ROLE | API Function | Cryptographic Keys & CSPs | Access |
|---|---|---|---|---|
| Module Initialization | CO | pds_crypt_init | Firmware Integrity ECDSA Public Key | Read, Execute |
| Self-Test /Show Status | CO, User | pds_crypt_init | Firmware Integrity ECDSA Public Key | Read, Execute |
| Module Exit | CO, User | pds_crypt_exit | Volume Keys | Write, Zeroize |
| Register Key | CO, User | pds_crypt_register_key | Volume Keys | Write |
| Zeroize | CO, User | pds_crypt_deregister_key | Volume Keys | Write, Zeroize |
| Check Registered Key | CO, User | pds_crypt_check_registered_key | Volume Keys | Read |
| Encrypt | CO, User | pds_crypt_encrypt | Volume Keys | Read, Execute |
| Decrypt | CO, User | pds_crypt_decrypt | Volume Keys | Read, Execute |

Exhibit 7 – *Services Authorized for Roles, Access Rights within Services (FIPS 140-2 Table C3, Table C4)*

# Algorithms

## Approved Algorithms

The Pavilion Cryptographic Module is designed to continually operate in a FIPS approved mode of operation. The Pavilion Cryptographic Module supports the following FIPS approved cryptographic algorithms:

| CAVP Cert | Algorithm | Standard | Mode / Method |
|---|---|---|---|
| A758 | SHA | 180-4 | SHA-256 |
| A759 | AES | FIPS 197, SP 800-38E, SP 800-38A | CBC[1](Encrypt/Decrypt 128 and 256), XTS(Encrypt/Decrypt 128 and 256) |
| A757 | ECDSA | FIPS 186-4 | SigVer P-256 SHA-256 |

Exhibit 8 – *Table of Approved Algorithms*

# Mitigation of Other Attacks Policy

The module is not designed to mitigate against attacks which are outside the scope of FIPS 140-2.

| Other Attacks | Mitigation Mechanism | Specific Limitations |
|---|---|---|
| N/A | N/A | N/A |

Exhibit 9 – *Mitigation of Other Attacks*

[1] This algorithm is only used as a pre-requisite to AES-XTS and is not a standalone algorithm.

## Physical Security

The module is a software-hybrid module that operates on a multi-chip standalone platform which conforms to the level 1 requirements for physical security. The hardware portion of the cryptographic module is a production grade component. A production grade enclosure completely surrounds the cryptographic module.

| Physical Security Mechanisms | Recommended Frequency of Inspection | Inspection Details and Guidance |
|---|---|---|
| N/A | N/A | N/A |

Exhibit 10 – *Inspection/Testing of Physical Security Mechanisms*

## Electromagnetic Interference/ Electromagnetic Compatibility (EMI/EMC)

The software runs in a platform that conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A.

## Operational Environment

As per FIPS 140-2, the Pavilion Cryptographic Module contains an operational environment that is modifiable, and meets the requirements for a level 1 software-hybrid module. The module is tested on the following operational environment:

- Oracle Linux 8.0 with Intel Xeon D-1548 with AES-NI Enabled

The module was tested on the following hardware:

- Pavilion RF140 Hyperparallel Flash Array; Software Version 2.7.0.0
- 10 IO Controller Cards contained within the Pavilion RF140 Hyperparallell Flash Array Chassis (2 independent IO controller nodes per card); Software Version 2.7.0.0


Exhibit 10 – *Intel Xeon D-1548 CPU*


Exhibit 11 – *Pavilion Hyperparallell Flash Array*

Exhibit 12 – *IO Controller Card*

As per FIPS Implementation Guidance (Section G.5), the Pavilion Cryptographic Module will remain compliant in all operational environments for which the binary executable remains unchanged. The Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module if the specific operational environment is not listed on the validation certificate.

The cryptographic module is implemented in a server environment. The server application is the user of the

cryptographic module. The server application makes the calls to the cryptographic module. Therefore, the server application is the single user of the cryptographic module, even when the server application is serving multiple clients.