Pure Storage, Inc.

Purity Encryption Module

FIPS 140-3 Cryptographic Module Non-Proprietary Security Policy

Version: 1.0

October 2024

Pure Storage, Inc.
650 Castro Street, Suite #260
Mountain View, CA 94041
800-379-7873

Table of Contents

List of Tables

1. General Information

This document defines the Security Policy for the Purity Encryption Module, hereafter denoted the Module. The Module is a multi-chip standalone software module (within the FlashArray product) and is run on a General Purpose Computer (GPC) with a modifiable operational environment. The Module meets FIPS 140-3 overall Level 1 requirements.

1.1 Security Levels

The Overall Security rating of the module is Level 1. Table 1 contains the security levels for all areas.

ISO/IEC 24759 Section 6.	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	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 Attack	N/A

Table 1 - Security Levels

2. Cryptographic Module Specification

Purity Encryption Module is a standalone cryptographic module for the Purity Operating Environment for FlashArray (Purity//FA). Purity//FA powers Pure Storage's FlashArray family of products which provide economical all-flash storage. Purity Encryption Module enables FlashArray to support always-on, inline encryption of data with an internal key management scheme that requires no user intervention.

The cryptographic module is defined as a software module.
The Module is intended for use by US Federal agencies and other markets that require FIPS 140-3 validated Data Storage.

2.1 Module Information

The validated module name is “Purity Encryption Module”, and the current version is output by the module as “FA-1.5”. The identifier “FA” is used as a module identifier.

2.2 Mode of Operation

The module supports a single Approved mode of operation. The Approved mode is enabled upon successful start-up of the module.

There are no specific initialization steps required for start-up of the module beyond powering on the FlashArray product.

The module does not support a non-approved mode of operation.

2.3 Operational Environments

Table 2 lists the operational environments the module was tested on.

#	Operating System	Hardware Platform	Processor(s)	PAA/PAI
1	Purity OS 6.4	FlashArray X20R3	Intel® Xeon® Silver 4210R	AES-NI
2	Purity OS 6.4	FlashArray X20R3	Intel® Xeon® Silver 4210R	None

Table 2 - Tested Operational Environments

Table 3 lists the operational environments that the vendor affirms can be used by the module. No claim is made as to the correct operation of the module or the security strengths of the generated keys when ported to an OE which is not listed on the validation certificate.

#	Operating System	Hardware Platform
1.	Purity OS 6.4	FlashArray X20 R4 with Intel® Xeon® Silver 4410Y
2.		FlashArray X50 R4 with Intel® Xeon® Silver 4410Y
3.		FlashArray X70 R4 with Intel® Xeon® Gold 5416S
4.		FlashArray X90 R4 with Intel® Xeon® Gold 5418N
5.		FlashArray C20 R4 with Intel® Xeon® Silver 4410Y
6.		FlashArray C50 R4 with Intel® Xeon® Silver 4410Y
7.		FlashArray C70 R4 with Intel® Xeon® Gold 5416S
8.		FlashArray C90 R4 with Intel® Xeon® Gold 5418N
9.		FlashArray C10 R3 with Intel® Xeon® Silver 4208
10.		FlashArray X50 R3 with Intel® Xeon® Silver 4214Y
11.		FlashArray X70 R3 with Intel® Xeon® Silver 6230
12.		FlashArray X90 R3 with Intel® Xeon® Silver 6252
13.		FlashArray C60 R3 with Intel® Xeon® Gold 6230
14.		FlashArray C40 R3 with Intel® Xeon® Silver 4210R
15.		FlashArray XL130 with Intel® Xeon® Gold 6338
16.		FlashArray XL170 with Intel® Xeon® Platinum 8368
17.	Purity OS 6.5	FlashArray X20 R4 with Intel® Xeon® Silver 4410Y
18.		FlashArray X50 R4 with Intel® Xeon® Silver 4410Y
19.		FlashArray X70 R4 with Intel® Xeon® Gold 5416S
20.		FlashArray X90 R4 with Intel® Xeon® Gold 5418N
21.		FlashArray C20 R4 with Intel® Xeon® Silver 4410Y
22.		FlashArray C50 R4 with Intel® Xeon® Silver 4410Y
23.		FlashArray C70 R4 with Intel® Xeon® Gold 5416S
24.		FlashArray C90 R4 with Intel® Xeon® Gold 5418N
25.		FlashArray C10 R3 with Intel® Xeon® Silver 4208
26.		FlashArray X20 R3 with Intel® Xeon® Silver 4210R
27.		FlashArray X50 R3 with Intel® Xeon® Silver 4214Y
28.		FlashArray X70 R3 with Intel® Xeon® Silver 6230
29.		FlashArray X90 R3 with Intel® Xeon® Silver 6252
30.		FlashArray C60 R3 with Intel® Xeon® Gold 6230

31.		FlashArray C40 R3 with Intel® Xeon® Silver 4210R
32.		FlashArray XL130 with Intel® Xeon® Gold 6338
33.		FlashArray XL170 with Intel® Xeon® Platinum 8368
34.	Purity OS 6.6	FlashArray X20 R4 with Intel® Xeon® Silver 4410Y
35.		FlashArray X50 R4 with Intel® Xeon® Silver 4410Y
36.		FlashArray X70 R4 with Intel® Xeon® Gold 5416S
37.		FlashArray X90 R4 with Intel® Xeon® Gold 5418N
38.		FlashArray C20 R4 with Intel® Xeon® Silver 4410Y
39.		FlashArray C50 R4 with Intel® Xeon® Silver 4410Y
40.		FlashArray C70 R4 with Intel® Xeon® Gold 5416S
41.		FlashArray C90 R4 with Intel® Xeon® Gold 5418N
42.		FlashArray C10 R3 with Intel® Xeon® Silver 4208
43.		FlashArray X20 R3 with Intel® Xeon® Silver 4210R
44.		FlashArray X50 R3 with Intel® Xeon® Silver 4214Y
45.		FlashArray X70 R3 with Intel® Xeon® Silver 6230
46.		FlashArray X90 R3 with Intel® Xeon® Silver 6252
47.		FlashArray C60 R3 with Intel® Xeon® Gold 6230
48.		FlashArray C40 R3 with Intel® Xeon® Silver 4210R
49.		FlashArray XL130 with Intel® Xeon® Gold 6338
50.		FlashArray XL170 with Intel® Xeon® Platinum 8368

Table 3 - Vendor Affirmed Operational Environments

2.4 Cryptographic Functionality

Table 4 below summarizes the approved algorithms supported by the module.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A4396	AES [FIPS 197] [SP 800-38A]	ECB, CTR	Key length: 128, 256 bits Key strength: 128, 256 bits	Symmetric Encryption and Decryption

A4396	AES [FIPS 197] [SP 800-38F]	KW	Key length: 128, 256 bits Key strength: 128, 256 bits	Key Wrapping and Unwrapping
A4396	KTS (AES) [SP 800-38F]	AES-KW	Key length: 128, 256 bits Key strength: 128, 256 bits	Key Wrapping and Unwrapping
A4396	CTR_DRBG [SP 800-90Arev1]	AES-256 Derivation Function Enabled Prediction Resistance: Yes	Key Strength: 256 bits	Random Number Generation
A4396	HMAC [FIPS 198-1]	SHA-256	Key length: 512 bits Key strength: 512 bits	Keyed Hash Verification
A4396	SHS [FIPS 180-4]	SHA-256	N/A	Message Digest

Table 4 - Approved Algorithms

Table 5 contains the non-approved algorithms allowed in the approved mode of operation with no security claimed.

Algorithm	Caveat	Use / Function
CRC32	No security claimed. Does not affect the operation of the approved algorithm implementation.	Used as an optional checksum on encrypted data when done during the AES CTR Decrypt service. The operation is separate from the execution of the AES CTR algorithms.

Table 5 - Non-Approved Algorithms Allowed in Approved mode of operation with No Security Claimed

There are no other non-approved algorithms allowed in the approved mode of operation.

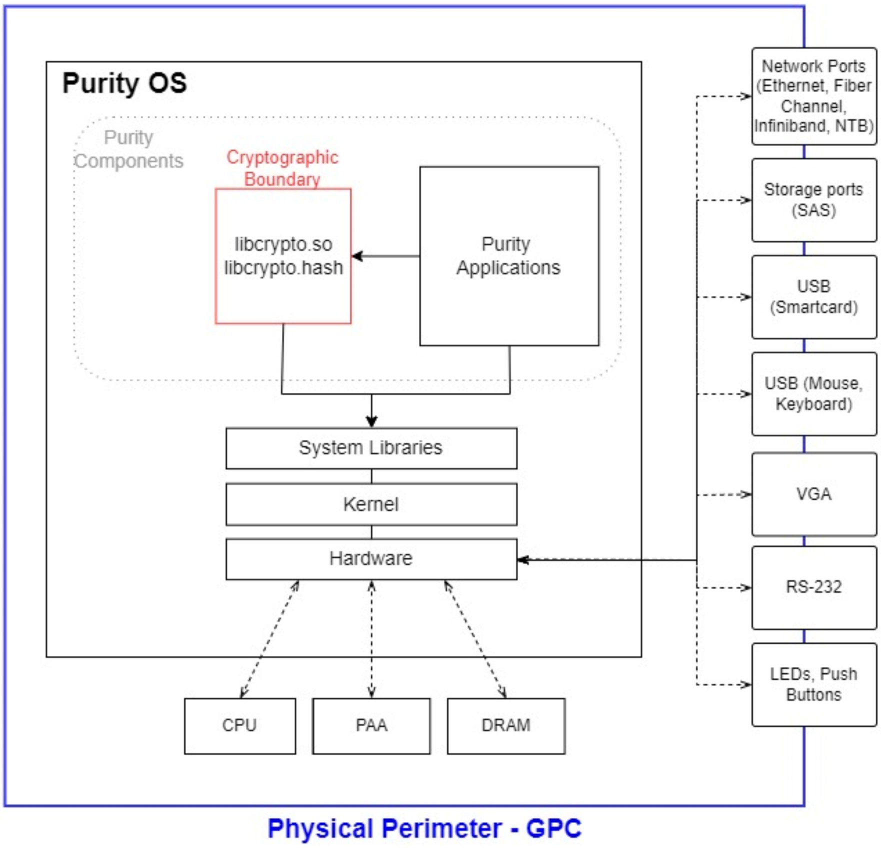
2.5 Cryptographic Module Boundary

The cryptographic boundary of the cryptographic module encompasses:

- libcrypto.so - the dynamically linked library libcrypto.so
- libcrypto.hash - the configuration file containing the module integrity code.

The Tested Operational Environment's Physical Perimeter (TOEPP) is defined as the physical General Purpose Computer (GPC) host platform on which the module is installed.

A block diagram depicting the physical and cryptographic boundaries is shown in the figure below:



3. Cryptographic Module Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-3 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs. The underlying logical interfaces of the module are the C++ language APIs.

The module supports four logical interfaces: Data Input, Data Output, Control Input and Status Output. It does not support a Control Output interface. Table 6 defines the logical interfaces.

Physical port	Logical Interface	Data that passes over port/interface
N/A	Data Input	The data read from memory area(s) provided to the invoked API functions via parameters that point to the memory area(s).
N/A	Data Output	The data written to memory area(s) provided to the invoked API functions via parameters that point to the memory area(s).
N/A	Control Input	The API function invoked, and API function parameters designated as control inputs.
N/A	Control Output	N/A
N/A	Status Output	The return value of the invoked API function.

Table 6 - Interfaces

4. Roles, Services, and Authentication

4.1 Roles

The module supports the Crypto Officer role, which is assumed implicitly by the operator of the module (the calling application) for all module services. No authentication mechanisms are provided to assume the Crypto Officer role. The module does not support concurrent operators and does not authenticate the Crypto Officer role. Furthermore, it does not support a maintenance role and/or bypass capability.

Table 7 below contains the services associated with available role.

Role	Service	Input	Output
Crypto Officer	AES encrypt	Key, IV, plaintext	Ciphertext
	AES decrypt	Key, IV, Ciphertext	Plaintext
	AES key wrap	Wrapping key, plaintext key	Wrapped key
	AES key unwrap	Wrapping key, wrapped key	Plaintext key
	Random Number Generation	-	64-bit random value
	Import key	Key data, type of key	-
	Export key	-	Key data
	Set entropy source	Function pointer	-
	Perform self-tests	Manual power cycle	Pass/fail
	Zeroisation	Manual power cycle	-
	Show status	-	Return code
	Show version	-	Name and version information

Table 7 - Roles, Service Commands, Input and Output

4.2 Services

Table 8 below lists the services that can be used in the approved mode of operation with corresponding input and output. The abbreviations of the access rights to keys and SSPs have the following interpretation.

G = Generate: The module generates or derives the SSP.
R = Read: The SSP is read from the module (e.g., the SSP is output).
W = Write: The SSP is updated, imported, or written to the module.
E = Execute: The module uses the SSP in performing a cryptographic operation.
Z = Zeroise: The module zeroises the SSP.
N**/A**= The service does not access any SSP during its operation

Name	Description	Approved Security Functions	SSPs	Role	SSP Access	Indicator
AES encrypt	Perform AES encryption	AES-ECB, AES-CTR	AES key	CO	W,E	Success/ Failure
AES decrypt	Perform AES decryption	AES-ECB, AES-CTR	AES key	CO	W,E	Success/ Failure
AES key wrap	Perform AES key wrap	AES-KW	AES Key Wrapping Key, AES Key	CO	W,E	Success/ Failure
AES key unwrap	Perform AES key unwrap	AES-KW	AES Key-Wrapping Key, AES Key	CO	W,E	Success/ Failure
Random Number Generation	Call random number generator	CTR_DRBG	Entropy Input, Seed, ‘V’ value, ‘Key’ value	CO	G,E,Z	Success/ Failure
Import key	Import key data into a key structure	n/a	AES key	CO	W	Success/ Failure
Export key	Export data from a key structure	n/a	AES key	CO	R	Success/ Failure
Set entropy source	Specifies callback function for entropy	N/A	n/a	CO		Success/ Failure
Perform self-tests	Power-cycling the host device.	AES, AES-KW, SHA-256, HMAC-SHA-256 DRBG		CO	E	Success/ Failure
Zeroisation	Power off/ cycle the host device	None	All	CO	Z	None

Show status	Return code for each API call	None	N/A	CO	None	None
Show version	Display the version of the module	N/A	None	CO	None	API invocation

Table 8 - Approved Services

5. Software/Firmware Security

5.1 Integrity Techniques

The Purity module is a single, shared object, binary component that is in an Executable and Linkable Format (ELF). A software integrity test is performed on this component using HMAC-SHA-256, which is implemented in the module. The integrity test succeeds if the computed integrity value is equal to the expected integrity value, loaded from the module's configuration file. If the integrity test fails, the module enters the Error State and the module becomes inoperable.

5.2 On-Demand Integrity Test

The integrity test is performed as part of the Pre-Operational Self-Tests, which are executed on load of the shared library. It can be also invoked by on demand by power-cycling the module.

6. Operational Environment

The Module is designated as a modifiable operational environment under the FIPS 140-3 definitions. The operational environment is the Purity Operating System for FlashArray 6.4, which is based on Ubuntu Linux. The operational environment implicitly enforces a single mode of operation by managing process memory of the module and ensuring each calling process is logically separated and protected.

No rules, settings or restrictions to the operational environment apply for operation of the module.

7. Physical Security

The FIPS 140-3 physical security requirements do not apply to the Purity module.

8. Non-invasive Security

The requirements of this area are not applicable to the module. This is not currently required by FIPS 140-3.

9. Sensitive Security Parameter Management

9.1 Keys and SSPs

Table 9 summarises the key Sensitive Security Parameters (SSPs) that are used by cryptographic services implemented in the module:

Key/SSP Name / Type	Strength	Security Function (and Cert #)	Gener-ation	Import / Export	Estab-lishment	Storage	Zero-isation	Use
AES Key (CSP)	128 and 256 bits	AES-ECB, AES-CTR [Cert A4396]	-	Imported or exported via API input or output parameters (Manual Distribution / Electronic Entry)	-	N/A: No persistent storage.	On power down	Symmetric Encryption and Decryption
AES Key-wrapping Key (CSP)	128 and 256 bits	AES-KW [Cert A4396]	-	Imported or exported via API input or output parameters (Manual Distribution / Electronic Entry)	-	N/A: No persistent storage.	On power down	Key Transport
DRBG Entropy Input (CSP)	256 bits	DRBG [Cert A4396]	Gathered from entropy source	-	-	N/A: No persistent storage.	On power down	DRBG entropy material
DRBG seed (CSP)	256 bits	DRBG [Cert A4396]	Generated internally	-	-	N/A: No persistent storage.	On power down	DRBG state value
DRBG ‘V’ value (CSP)	256 bits	DRBG [Cert A4396]	Generated internally	-	-	N/A: No persistent storage.	On power down	DRBG state value
DRBG ‘Key’ value (CSP)	256 bits	DRBG [Cert A4396]	Generated internally	-	-	N/A: No persistent storage.	On power down	DRBG state value
Software integrity HMAC key (Non-SSP)	512 bits	HMAC [Cert A4396]	-	-	-	Stored in the module binary.	N/A	Self-test

Table 9 - Keys and SSPs

9.2 DRBG and Entropy Sources

The module employs a single DRBG instance for the purpose of supplying random numbers to the calling application via the module's service random number generation service. This DRBG is used for this service only and is not used by the module for any other purpose. This module DRBG is CTR_DRBG with security strength of 256 bits.

The module DRBG is seeded from the module's entropy source, which is set by the calling application via a callback function as input to the applicable service API. The calling application and the supplied entropy source are external to the module boundary. It is a requirement of the FlashArray product that the entropy source configured for the module provides full entropy. As such, the 256-bits of data loaded from the entropy source to seed the module DRBG provides 256-bits of entropy.

Entropy Sources	Minimum number of bits of entropy	Details
Application specified entropy source	256-bits	External entropy source set by the calling application via a callback function. Required to provide full entropy.

Table 10 - Non-Deterministic Random Number Generation Specification

10. Self-tests

This section specifies the pre-operational self-tests and conditional self-tests performed by the module. They include the software integrity test and the cryptographic algorithm self-tests. These tests ensure that the module is not corrupted and the algorithms function as expected.

All self-tests are executed automatically when the module is loaded into memory before the module transitions to the operational state. The services of the module are not available prior to the completion of the self-tests. Successful completion of self-tests is indicated by a status message and passing control to the calling application.

Failure of any self-test causes the module to enter the Error State, which is indicated by an error message. In this error state, the module is not operational, and no services are available. This is the only error state supported by the module.

The module permits operators to initiate the self-tests on demand by power cycling the system.

10.1 Pre-operational Self-Tests

The module performs the following pre-operational test:

- Software integrity test (using HMAC SHA256)

10.2 Conditional Self-Tests

The module performs the following conditional Cryptographic Algorithm Self-Tests (CASTs) via Known Answer Tests (KAT):

- AES CTR Encrypt and Decrypt KATs (128, 256 bits)
- AES ECB Encrypt and Decrypt KATs (128, 256 bits)
- AES KW Wrap and Unwrap KATs (256 bits)
- CTR-DRBG KATs for Instantiate, Generate, Reseed
- SHA-256 KAT
- HMAC-SHA256 KAT (512 key bits)

All CASTs are triggered by the module’s initial load sequence.

11. Life-cycle assurance

11.1 Delivery and Operation

The module is built into Purity OS. There is no standalone delivery of the module as a software library.

The vendor’s internal development process guarantees that the correct version of module is distributed with the intended device.

The module does not have any specific maintenance requirements.

11.2 Crypto Officer Guidance

There is only one Approved mode of operation. Crypto Officer Role Guidance is provided by the API documentation provided by the module’s header files.

12. Mitigation of other attacks

The module does not claim mitigation of other attacks.