# Google, LLC.

# Non-Proprietary FIPS 140-3 Security Policy for

# Tensor G2 UFS Inline Storage Encryption Cryptographic Module

Software Version: 1.2.0

Hardware Version: 4.1.0

Documentation Version : 1.1
Last Update: June 18, 2024

# Table of Contents



# List of Figures



# List of Tables

## 1. General

This document is the non-proprietary FIPS 140-3 Security Policy for the Tensor G2 UFS Inline Storage Encryption Cryptographic Module. It contains a specification of the 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 a Security Level 1 Software-Hybrid cryptographic module. This security policy is for the validation of the Tensor G2 UFS Inline Storage Encryption Cryptographic Module.

In this document, the terms "Tensor G2 UFS Inline Storage Encryption Cryptographic Module", "cryptographic module" or "module" are used interchangeably to refer to the Tensor G2 UFS Inline Storage Encryption Cryptographic Module with Software version 1.2.0 and Hardware version 4.1.0.

*Table 1-1 Security Levels*

| 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 | 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** | **Security Level 1** | |

## 2. Cryptographic Module Specification

The module is a multi-chip standalone Software-Hybrid module designed for on-the-fly hardware encryption for a flash storage device. The module's cryptographic boundary includes the following components:

- UFS Pixel FIPS CMVP Module (Module's Software Component, version 1.2.0)
- UFS ISE (Module's Hardware Component, version 4.1.0)

The UFS Pixel FIPS CMVP Module (hereafter referred to as ISE Driver) is the software component of the cryptographic module running in the Linux Kernel, which calls the cryptographic algorithms for the module's pre-operational self-tests, and also sets the FIPS status of the entire cryptographic module once the pre-operational self-test is completed successfully.

The UFS ISE is the hardware component of the cryptographic module that supports AES-XTS encryption and decryption. This hardware resides in the Google Tensor G2 processor, located between the device's DRAM and the Flash Storage Device so it can provide inline encryption/decryption while maintaining device performance (such as responsiveness and power consumption). Please see Figures 1, 2 and 3 below for more information.

The module is specified in the following table.

| Component | Type | Version |
|---|---|---|
| UFS Pixel FIPS CMVP Module | Software | 1.2.0 |
| UFS ISE | Hardware | 4.1.0 |

The module has been tested on the following platforms.

*Table 2-1 Tested Operational Environments*

| Operating Systems | Tested Platform Hardware Versions | Processor on the Tested Platforms | PAA\Acceleration |
|---|---|---|---|
| Linux Kernel 5.10 | Google Pixel 7 | Google Tensor G2 | With PAA |

The table below lists approved cryptographic algorithms employed by the module:

*Table 2-2 Approved Algorithms*

| CAVP Cert | Algorithm and Standard | Mode / Method | Description / Key Size(s) / Key strength(s) | Use / Function |
|---|---|---|---|---|
| #A2937 | AES [FIPS 197; SP 800-38E] | AES-XTS | 256 bits | Symmetric Encryption/Decryption |
| #A2938 | HMAC [FIPS 198-1] | HMAC-SHA2-256 | At least 112 bits | Software integrity test |
| #A2938 | SHA [FIPS 180-4] | SHA2-256 | N/A | Pre-requisite algorithm of HMAC-SHA2-256 used for software integrity test |

Notes:

- Not all CAVP tested modes of the algorithms are used in this module
- According to SP 800-38E, the AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in SP800-38E. In addition, the length of a single data unit encrypted with the XTS-AES shall not exceed $2^{20}$ AES blocks, that is, 16 MiB of data. In addition, to meet the requirement in FIPS 140-3 IG C.I, the module implements a check to ensure that the two AES keys (Key 1 and Key 2) used in XTS-AES algorithm are not identical.

### Mode of Operation

The module always runs in the Approved Mode of Operation and does not implement any Non-Approved Security Functions. The module doesn't support degraded operational mode.

### Block Diagram

In the following diagram, the bidirectional arrows depict the flow of the status, control and data within the device's Tested Operational Environment's Physical Perimeter (TOEPP). The TOEPP in the block diagram contains the module (the blue dotted region). The operations within the module's cryptographic boundary (the blue dotted region) use the cipher from the UFS ISE (Module's hardware component) that is included in the module's cryptographic boundary. The UFS Pixel FIPS CMVP Module (Module's software component) is only used at module's initialization to perform the pre-operational self-tests. The module's executable file is ufs-pixel-fips140.ko kernel loadable module.

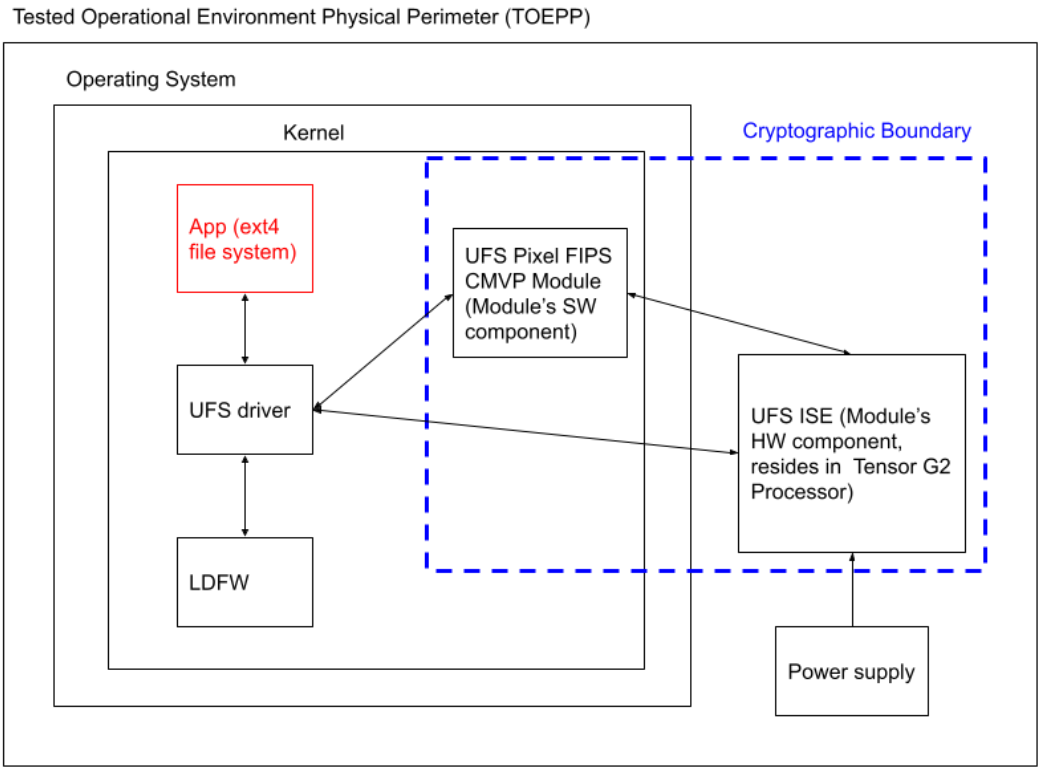


*Figure 1. Module's Block Diagram*

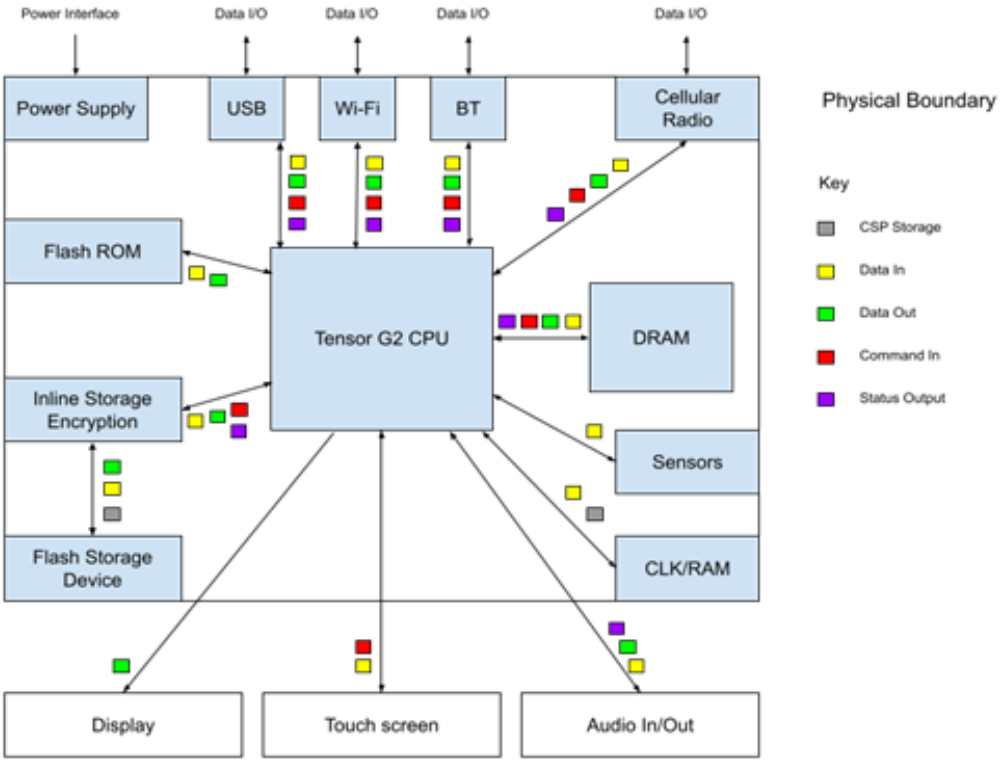


*Figure 2. Tested Platform Physical Perimeter*

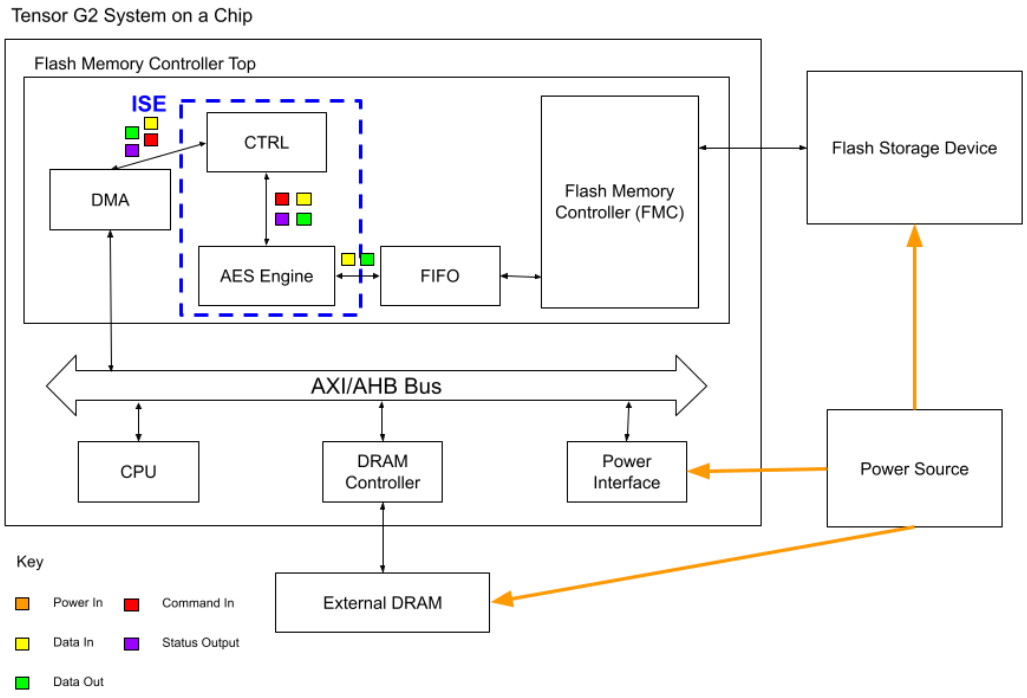


*Figure 3. Module's Hardware Block Diagram*

## 3. Cryptographic Module Interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in Table 2-1. The module provides its logical interfaces via Application Programming Interface (API) calls. The logical interfaces provided by the module are mapped onto the FIPS 140-3 interfaces (data input, data output, control input, control output and status output) as follows.

*Table 3-1 Ports and Interfaces*

| Physical Port | Logical Interface | Data that passes over port/interface |
|---|---|---|
| DMA FIFO Interface, DMA CTRL Interface | Data Input Interface | Arguments for an API call that provide the data to be used or processed by the module. |
| DMA FIFO Interface, DMA CTRL Interface | Data Output Interface | Output data returned to calling function |
| DMA CTRL Interface | Control Input Interface | Arguments for an API call used to control and configure module operation. |
| DMA CTRL Interface | Status Output Interface | Return values from the Module's API used to obtain information on the status of the module. The Status Output Interface also includes the log file where the module messages are output. |
| N/A | Control Output Interface | N/A |
| Power Interface | N/A | Module's hardware component power supply |

## 4. Roles, Services and Authentication

The module supports Crypto Officer (CO). The cryptographic module does not provide any authentication methods. The module does not allow concurrent operators. The Crypto Officer is implicitly assumed based on the service requested.

The module provides the following services to the Crypto Officer.

*Table 4-1 Roles, Service Commands, Input and Output*

| Role | Service | Input | Output |
|---|---|---|---|
| **Module's Hardware Component** | | | |
| Crypto Officer (CO) | AES-XTS encryption and decryption | Key and message (plaintext message for Encryption or cipher text for decryption) | Encrypted message or Decrypted message |
| **Module's Software Component** | | | |
| Crypto Officer (CO) | Show Version | API Command to get module's version | Module's ID and component's versions (SW and HW) |
| Crypto Officer (CO) | Perform Self-Tests | On-Demand Self-Test (Power cycling) | Pass/Fail status<br>Note: Return value of 1 for success; Failure results in panic to kernel |
| Crypto Officer (CO) | Show Status | API Command to check the status | Module's operational status |
| Crypto Officer (CO) | Perform Zeroization | API Command to zeroize all SSPs | Zeroized and released memory space |

Table 4-2 defines the relationship between access to SSPs and the different module services. The modes of access shown in the table are defined as:

- 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 = Zeroize: The module zeroizes the SSP.

*Table 4-2 Approved Services*

| Service | Description | Approved Security Functions | Keys and/or SSPs | Roles | Access Rights to SSPs | Indicator |
|---|---|---|---|---|---|---|
| AES-XTS encryption and decryption | Perform Symmetric Encryption and Decryption for Disk protection | AES-XTS | AES-XTS Key | CO | W, E | Symmetric encryption and decryption completion status |
| Show Version | Show module's ID and component's versions (SW and HW) | N/A | N/A | CO | N/A | None |
| Perform Self-Tests | Run Pre-operational Self-Test and Conditional Algorithm Self-Tests | N/A | Software Integrity Key (not SSP) | CO | E | None |

| Service | Description | Approved Security Functions | Keys and/or SSPs | Roles | Access Rights to SSPs | Indicator |
|---|---|---|---|---|---|---|
| Show Status | Show module's current status | N/A | N/A | CO | N/A | None |
| Perform Zeroization | Zeroize the SSPs stored in the module | N/A | AES-XTS Key | CO | Z | None |

Note: With regard to the Indicator defined in FIPS 140-3 standard, as the Module is always operated in the approved mode (without the Operator's configuration), the service successful completion status will be functioning as the Indicator.

## 5. Software/Firmware Security

### Integrity Techniques

To ensure software security, a software integrity test is performed on the runtime image of the module. The HMAC-SHA2-256 (HMAC Cert. #A2938) implemented in the module is used as an approved algorithm for the integrity test. If the test fails, the module enters an error state where no cryptographic services are provided, and data output is prohibited. The module is provided in the form of binary executable code.

### On-Demand Integrity Test

Software Integrity test is performed as part of the Pre-Operational Self-Tests. It is automatically executed at power-on. Thus, it can be invoked by rebooting the tested platform.

## 6. Operational Environment

The module operates in a modifiable operational environment per FIPS 140-3 Security Level 1 Specifications. The operating system is restricted to a single operator mode of operation. The procurement, build and configuring procedure are controlled. The module is installed into a commercial production grade mobile device. The external application that makes calls to the cryptographic module is the single instance of the cryptographic module, even when the application is serving multiple clients.

## 7. 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. All disjoint components of the module are entirely contained within the production-grade enclosure of the host platform, which blocks physical access to the module. The tested platform (mobile device) shall comprise production grade components with standard passivation (a sealing coat applied over the chip circuitry to protect it against environmental and other physical damage) and a production grade enclosure that completely surrounds the cryptographic module.

## 8. Non-invasive Security

The module does not support Non-invasive Security. Thus, the security requirements from Section Non-invasive Security in FIPS 140-3 are not applicable.

## 9. Sensitive Security Parameter Management

*Table 9-1 SSPs*

| Key/SSP/ Name Type | Strength | Security Function and Cert. | Generation | Import/ Export | Establishment | Storage | Zeroization | Use & related keys |
|---|---|---|---|---|---|---|---|---|
| AES-XTS Key | 256 bits | AES-XTS<br><br>Cert # A2937 | N/A | Import: Entered via Module's API (plaintext)<br><br>Export: No | N/A | Temporarily stored in Module's hardware registers. | Power down the tested platform | Used for Symmetric Encryption and Decryption |

Notes:

**Key Generation**

The module does not provide any key generation service or perform any key generation for any of its Approved algorithms. Keys are instead provided by third party applications and stored in memory location outside of the cryptographic module boundary (i.e., within a DMA descriptor located in memory within the Tested Operational Environment's Physical Perimeter (TOEPP) of the tested platform). The module does not support any key establishment methods or asymmetric algorithms and hence no key generation services for them.

**Key Entry and Output**

The module does not support manual key entry or key output. SSP (AES-XTS Key is the only SSP) can only be exchanged via a DMA descriptor inside the TOEPP of the device. All SSPs are entered to module per the request from the module's calling application running on the same tested platform. Keys/SSPs are electronically entered into the module via Module's API in plaintext form. The Module doesn't output the SSPs.

**Key Storage**

The module does not provide persistent keys/SSPs storage. After the SSP (AES-XTS Key is the only SSP) is entered to the module via the Module's API, the module temporarily stores it in the Module's hardware register. The Module's hardware register is internal to the hardware module and is not shared with any external component (operating system or other hardware). No process other than the module itself can access the keys/SSPs in its memory.

**Key Zeroization**

All SSPs are zeroized when the system is powered down. Input and output interfaces are inhibited while zeroization is performed. The successful act of powering off the module serves as the implicit indicator of zeroization.

## 10. Self-Tests

When the module is loaded or instantiated (after being powered off, rebooted, etc.), the module runs pre-operational self-tests. The operating system is responsible for the initialization process and loading of the library. The module is designed with a default entry point (DEP) which ensures that the self-tests are initiated automatically when the module is loaded. Prior to the module providing any data output via the data output interface, the module would perform and pass the pre-operational self-tests. Following the successful pre-operational self-tests, the module would execute the Conditional Cryptographic Algorithm Self-tests (CASTs).

The self-test success or failure is output as a return value of the library load API call, which is functioning as the self-test status indicator. If one of the self-tests fails, the module transitions into an error state and outputs the error message via the module's status output interface. While the module is in the error state, all data through the data output interface and all cryptographic operations are disabled. The error state can only be cleared by reloading the module. All self-tests must be completed successfully before the module transitions to the operational state.

Below are the details of the self-tests conducted by the module.

### Pre-Operational Self-Tests

- Pre-Operational Software Integrity Self-Tests
  - HMAC-SHA2-256 KAT
  - Software Integrity Test (using HMAC-SHA2-256)

The module software integrity test parameters are configured at build time and then executed at runtime. As the module supports KASLR and is not loaded as a contiguous block, a series if begin/end addresses (__fips140_text_start/__fips140_text_end and __fips140_rodata_start/__fips140_rodata_end) are used to determine the code that must be checked. The HMAC value (ufs_pixel_fips_hmac_key) of that code is then calculated when building the ELF, storing the calculated digest (ufs_pixel_fips_hmac_expected) in the module.

The driver is a kernel loadable module included in the vendor image for the kernel and loaded once the ramdisk holding all the kernel modules has been unpacked into memory.

Once the module is loaded, the HMAC-SHA2-256 is calculated over the entire area specified as part of the build process. This is then compared to the stored value (ufs_pixel_fips_hmac_expected) from the build time calculation. A non-match will place the module into the error state.

The Module conducts HMAC-SHA2-256 KAT self-test before the integrity test is performed.

### Conditional Algorithm Self-Tests (CASTs)

- Conditional cryptographic algorithm tests
  - AES-XTS Encryption Know Answer Test (KAT)
  - AES-XTS Decryption KAT
  - HMAC-SHA2-256 KAT
  - SHA2-256 KAT

The module conducts the CASTs before the first operational use of the cryptographic algorithm.

### Periodic/On-Demand Self-Test

The module performs on-demand self-tests initiated by the operator, by powering off and powering the module back on. The full suite of self-tests is then executed. The same procedure may be employed by the operator to perform periodic self-tests.

## 11. Life-cycle Assurance

The module is enabled by default (and hence used automatically) as part of the device without any user configuration. The module always runs in the Approved Mode of Operation and does not implement any Non-Approved Security Functions. When the module is loaded or instantiated (after being powered off, rebooted, etc.), the module runs pre-operational self-tests without any operator intervention. The Module will be operated in an approved mode of operation when pre-operational self-tests have completed successfully.

The module is provided directly to solution developers and is not intended for direct download by the general public. The module is installed on an operating system (Linux kernel 5.10) specified in Section 2.

Additional Rules of Operation:

1. The module does not support concurrent operators.

2. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the module.

3. The end user of the operating system is also responsible for zeroizing SSPs via wipe/secure delete procedures

## 12. Mitigation of Other Attacks

The module does not support Mitigation of Other Attacks. Thus, the security requirements from Section Mitigation of Other Attacks in FIPS 140-3 are not applicable.