Google, LLC.

Titan-BPN Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.0
March 3rd, 2026

Prepared by:

www.lightshipsec.com

Table of Contents

1 General 5
- 1.1 Overview 5
- 1.2 Security Levels 5

2 Cryptographic Module Specification 6
- 2.1 Description 6
- 2.2 Tested and Vendor Affirmed Module Version and Identification 7
- 2.3 Excluded Components 8
- 2.4 Modes of Operation 8
- 2.5 Algorithms 8
- 2.6 Security Function Implementations 9
- 2.7 Algorithm Specific Information 10
- 2.8 RBG and Entropy 10
- 2.9 Key Generation 10
- 2.10 Key Establishment 10
- 2.11 Industry Protocols 10

3 Cryptographic Module Interfaces 11
- 3.1 Ports and Interfaces 11

4 Roles, Services, and Authentication 12
- 4.1 Authentication Methods 12
- 4.2 Roles 12
- 4.3 Approved Services 12
- 4.4 Non-Approved Services 14
- 4.5 External Software/Firmware Loaded 16

5 Software/Firmware Security 17
- 5.1 Integrity Techniques 17
- 5.2 Initiate on Demand 17
- 5.3 Additional Information 17

6 Operational Environment 19
- 6.1 Operational Environment Type and Requirements 19

7 Physical Security 20
- 7.1 Mechanisms and Actions Required 20

8 Non-Invasive Security 21

9 Sensitive Security Parameters Management 22
- 9.1 Storage Areas 22

9.2 SSP Input-Output Methods 22
9.3 SSP Zeroization Methods 22
9.4 SSPs 22
9.5 Transitions 23
10 Self-Tests 24
10.1 Pre-Operational Self-Tests 24
10.2 Conditional Self-Tests 24
10.3 Periodic Self-Test Information 25
10.4 Error States 26
11 Life-Cycle Assurance 27
11.1 Installation, Initialization, and Startup Procedures 27
11.2 Administrator Guidance 27
11.3 Non-Administrator Guidance 27
12 Mitigation of Other Attacks 28

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Hardware 7
Table 3: Modes List and Description 8
Table 4: Approved Algorithms 9
Table 5: Non-Approved, Not Allowed Algorithms 9
Table 6: Security Function Implementations 10
Table 7: Entropy Certificates 10
Table 8: Entropy Sources 10
Table 9: Ports and Interfaces 11
Table 10: Roles 12
Table 11: Approved Services 14
Table 12: Non-Approved Services 16
Table 13: Mechanisms and Actions Required 20
Table 14: Storage Areas 22
Table 15: SSP Zeroization Methods 22
Table 16: SSP Table 1 23
Table 17: SSP Table 2 23
Table 18: Pre-Operational Self-Tests 24
Table 19: Conditional Self-Tests 25
Table 20: Pre-Operational Periodic Information 25
Table 21: Conditional Periodic Information 25
Table 22: Error States 26

List of Figures

Figure 1: Titan-BPN Cryptographic Module Block Diagram 6
Figure 2: Top of Titan-BPN Cryptographic Module 7
Figure 3: Bottom of Titan-BPN Cryptographic Module 7

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the Google, LLC. Titan-BPN Cryptographic Module (running firmware version gqfips-1.3), hereafter referred to as, “Titan-BPN” or “the module”. It contains the security rules under which the module must operate and describes how the module meets the requirements as specified in FIPS PUB 140-3 for an overall Security Level 1 cryptographic module.

1.2 Security Levels

The table below reflects the individual security areas of FIPS 140-3, as well as the Security Levels of those individual areas.

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 Level	1

Table 1: Security Levels

The Module has an overall security level of 1.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The module is a custom secure microcontroller. It can implement a variety of security, encryption, and cryptography protocols. The protocols are running on a secure processor on-chip, interfacing with a host using an API across a trusted SPI peripheral. It provides secure EEPROM Boot, using SPI pass-through technology that allows Titan-BPN to confirm authorship of Boot Code, ensuring code-signing before a code swap is completed.

Titan-BPN is embedded in Google, LLC.’s River Redux Cryptographic Module (FIPS 140-3 Cert. #5165). The River Redux Cryptographic Module utilizes Titan-BPN for entropy and to validate and perform the integrity test on its firmware.

Module Type: Hardware

Module Embodiment: SingleChip

Cryptographic Boundary:

The cryptographic boundary of the module is the outer perimeter of the chip, as shown in the figures below.

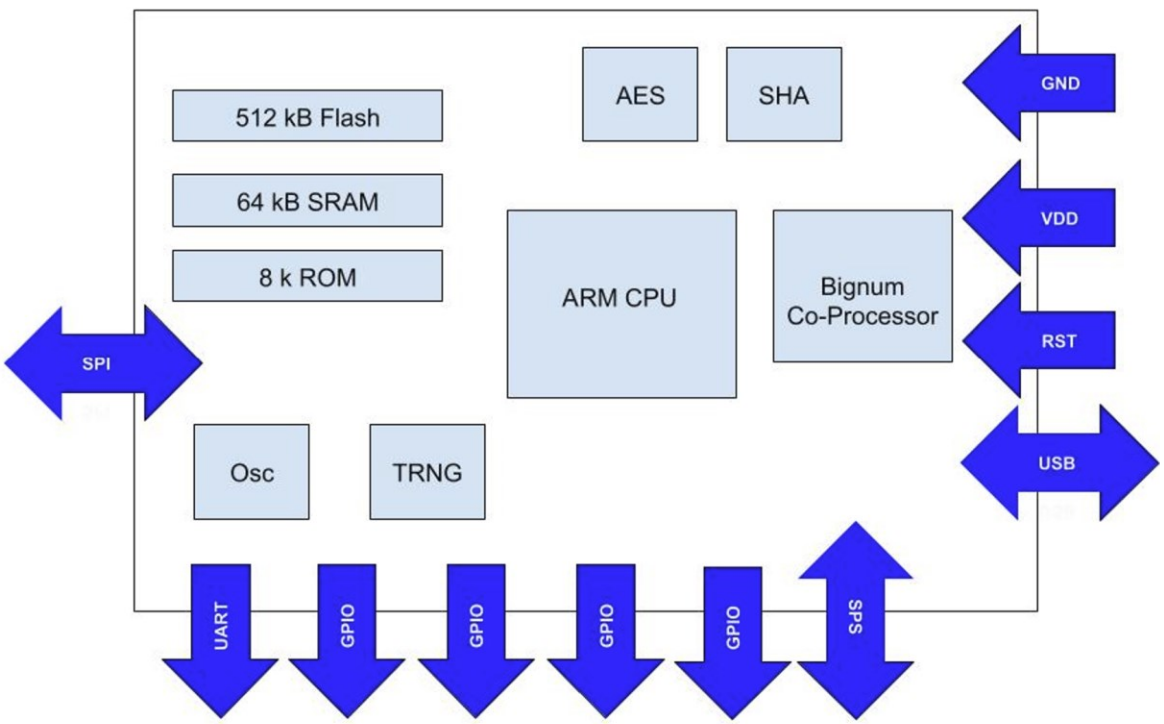

Figure 1: Titan-BPN Cryptographic Module Block Diagram

Figure 2: Top of Titan-BPN Cryptographic Module

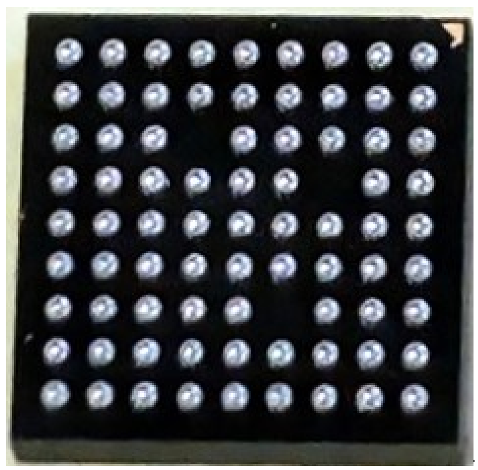

Figure 3: Bottom of Titan-BPN Cryptographic Module

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
Titan-BPN Cryptographic Module	H1B3P	gqfips-1.3	ARM SC300 processor, with Dcrypto Coprocessor	

Table 2: Tested Module Identification – Hardware

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

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:

The table below details the Modes of Operation supported by the module.

Mode Name	Description	Type	Status Indicator
Approved Mode	Operation mode where the module executes approved services	Approved	fips_indicator: "0"
Non-Approved Mode	Operation mode where the module executes non-approved services	Non-Approved	fips_indicator: "1"

Table 3: Modes List and Description

Mode Change Instructions and Status:

The Titan-BPN Cryptographic Module supports Approved and Non-Approved security functions. Approved services provide an explicit indicator when the service utilizes an approved cryptographic algorithm. In accordance with FIPS 140-3 IG 2.4.C, an operator of the module can unambiguously determine if and when an Approved security service is in use based on the “fips_indicator” return code returned to the module’s status output interface upon a request. A value of “0” indicates an Approved security service has been requested. A value of “1” indicates a non-Approved security service has been requested.

2.5 Algorithms

Approved Algorithms:

The table below lists all the Approved Algorithms supported by the module.

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-4)	A5419	Curve - P-256 Hash Algorithm - SHA2-256	FIPS 186-4
RSA SigVer (FIPS186-4)	A5419	Signature Type - PKCS 1.5 Modulo - 2048, 3072	FIPS 186-4
SHA2-256	A5419	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A5420	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

The module does not support any Vendor-Affirmed Algorithms in the Approved Mode of Operation.

N/A for this module.

Non-Approved, Allowed Algorithms:

The module does not support any Non-Approved, Allowed Algorithms in the Approved Mode of Operation.

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

The module does not support any Non-Approved, Allowed Algorithms with No Security Claimed in the Approved Mode of Operation.

N/A for this module.

Non-Approved, Not Allowed Algorithms:

The table below lists all the Non-Approved Algorithms that are not Allowed in the Approved Mode of Operation.

Name	Use and Function
HMAC-SHA2-256 (non-conformant)	Keyed-Hash
HKDF-SHA2-256 (non-conformant)	Hash Key Derivation Function
ECDSA P-256 (non-conformant)	Signature Generation, Signature Verification, Key Pair Generation
EC DH P-256 (non-conformant)	Key Establishment
AES-CTR (non-conformant)	Encryption, Decryption
HMAC DRBG (non-conformant)	Random Number Generation

Table 5: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

The table below lists the Security Function Implementations supported by the module.

Name	Type	Description	Properties	Algorithms
Titan-BPN Chip Message Digest	SHA	Hashing		SHA2-256: (A5420)
Titan-BPN Library Message Digest	SHA	Hashing		SHA2-256: (A5419)
Titan-BPN Library FW Updates Verification	DigSig-SigVer	Digital Signature Verification		ECDSA SigVer (FIPS186-4): (A5419) SHA2-256: (A5419)

Name	Type	Description	Properties	Algorithms
Titan-BPN Library EEPROM FW Verification	DigSig-SigVer	Digital Signature Verification		RSA SigVer (FIPS186-4): (A5419) SHA2-256: (A5419)
Titan-BPN TRNG Entropy Source	ENT-Cond ENT-ESV	Entropy Source		SHA2-256: (A5420)

Table 6: Security Function Implementations

2.7 Algorithm Specific Information

There is no algorithm specific information.

2.8 RBG and Entropy

The tables below detail the module ESV information.

Cert Number	Vendor Name
E187	Google, LLC.

Table 7: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Titan-BPN TRNG Entropy Source	Physical	Titan Chip H1B3P	256 bits	256 bits	SHA2-256 (A5420)

Table 8: Entropy Sources

The module implements a hardware-based True-Random Number Generator (TRNG). The TRNG is used to generate entropy which is output as service to the directly connected River Redux Cryptographic Module (FIPS 140-3 Cert. #5165).

Upon request the module's TRNG will output 256 bits blocks of entropy via the module's SPS interface to the embedding River Redux Cryptographic Module.

2.9 Key Generation

The module does not implement any approved key generation methods.

2.10 Key Establishment

The module does not implement any automated approved key establishment methods.

2.11 Industry Protocols

The module does not implement any industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The table below details the module Ports and Interfaces. The ports are pins on the single chip

Physical Port	Logical Interface(s)	Data That Passes
VDD	Power	Supply Voltage
RST	Control Input	Reset Signal
CLK	None	Not used
GND	Power	Ground
SPS	Data Input Data Output Control Input Control Output Status Output	SPI slave from NIC
SPI	Data Input Data Output	SPI master to external EEPROM
BOOTSTRAP (GPIO)	Control Input	Set to Bootstrap module during initialization. *This pin is not used by production NIC.
STRAPPING (GPIO)	Control Input	Bit containing profile Information
GOOD (GPIO)	Status Output	Status bit
UART TX	Status Output	Debug log
GPIO	None	Not used
NC	None	Not used
USB	Data Input Data Output Control Input Status Output	Direct connection to the host via USB

Table 9: Ports and Interfaces

The module contains an SPI master interface and an SPI slave interface. The master interface is used to initiate flash commands to the external EEPROM, and the slave interface is used to receive commands initiated from the NIC.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not support authentication for roles.

N/A for this module.

4.2 Roles

The module supports two roles that an operator may assume: Crypto Officer (CO) role and User role. Roles are assumed implicitly based on the service accessed. The table below lists the Roles supported by the module.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None
User	Role	User	None

Table 10: Roles

4.3 Approved Services

The table below lists all Approved Services supported by the module. The abbreviations of the access rights to 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.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Initialization	Module Initialization	N/A	None	None	None	Crypto Officer User
Show Version	Return the module version information	N/A	API command and parameters	Module Versioning Information	None	Crypto Officer
Show Status	Return the module status	N/A	API command and parameters	Module Status Information	None	Crypto Officer User
On-Demand Self-test	Initiate on-demand self-tests by power cycling the module	N/A	None	Pass or Fail status	None	Crypto Officer User
Zeroisation	Zeroise contents of	N/A	None	None	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	volatile memory					
Check Status of Partitions	Check Status of Staging Partition	N/A	API command and parameters	Success or Fail status	None	Crypto Officer User
Verify and Activate Staging Partition	Verify and activate a partition for use on the next boot. EEPROM Firmware Verification	fips_indicator: "0"	API command and parameters	Success or Fail status	Titan-BPN Library Message Digest Titan-BPN Library EEPROM FW Verification	Crypto Officer - Payload Key: G,E User - Payload Key: G,E
Check for Firmware Update	Check for available FW updates	N/A	API command and parameters	New FW present, No new FW present	None	Crypto Officer
Perform Firmware Update	Applies firmware updates	New Firmware Version	API command and parameters	Success or Fail status	Titan-BPN Library Message Digest Titan-BPN Library FW Updates Verification	Crypto Officer - EEPROM Firmware Verification Key: G,E - Downgrade Authorization Key: G,E
Provide Raw Entropy Output	Send River Redux raw entropy	fips_indicator: "0"	API command and parameters	Raw entropy	Titan-BPN TRNG Entropy Source	Crypto Officer User
Provide Whitened Entropy Output	Send River Redux whitened entropy	fips_indicator: "0"	API command and parameters	Whitened entropy	Titan-BPN Chip Message Digest Titan-BPN TRNG Entropy Source	Crypto Officer User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
SPI Read Request to Active Partition of External storage	Reads from the active partition on external storage	N/A	API command and parameters	Success or Fail status	None	Crypto Officer User
SPI Write request to Firmware Staging Region on External storage	Writes to the firmware staging region on external storage	N/A	API command and parameters	Success or Fail status	None	Crypto Officer User
Write request to SPI Staging Partition on External storage	Write data to the SPI staging area on external storage	N/A	API command and parameters	Success or Fail status	None	Crypto Officer User
RSA Signature Verification Operation	RSA Signature Verification	fips_indicator: "0"	API command and parameters	Success or Fail status	Titan-BPN Library Message Digest Titan-BPN Library EEPROM FW Verification	Crypto Officer User

Table 11: Approved Services

4.4 Non-Approved Services

The table below lists all Non-Approved Services supported by the module.

Name	Description	Algorithms	Role
AIM Generate Alpha Attestation Key	Alpha Identity Manager (AIM) generate Alpha Attestation Key from the TRNG	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant)	User
AIM Load and Verify Alpha Attestation Key	Load and Verify Alpha Attestation Key	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant)	User

Name	Description	Algorithms	Role
		ECDSA P-256 (non-conformant)	
AIM Load and Verify Alpha Attestation Key from CSR	Load and Verify Alpha Attestation Key Certificate Signing Request	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant)	User
AIM Retrieve Aplha Bind Key Counter Value	Retrieves the counter value associated with a specified Alpha Bind Key	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant)	User
AIM Generate Alpha Bind Key	Generates a new wrapped Alpha Bind Key	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant)	User
AIM Load Alpha Bind Key	Loads an Alpha Bind Key	HKDF-SHA2-256 (non-conformant) AES-CTR (non-conformant)	User
AIM Validate Alpha Bind Key	Performs validation of Alpha Bind Kind. Does not load the key	HKDF-SHA2-256 (non-conformant) AES-CTR (non-conformant)	User
AIM Get Version	Derives Alpha Attestation Key and binds it to the firmware release	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant)	User
AIM Get alias key certificate	Retrieves Alias key certificate from internal storage	ECDSA P-256 (non-conformant) HMAC DRBG (non-conformant)	User
AIM Get certificate signed with alias key for Alpha Attestation Key	Retrieves Public Key, Certificate signed with alias key for currently loaded Alpha Attestation Key	HMAC-SHA2-256 (non-conformant) HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant) HMAC DRBG (non-conformant)	User

Name	Description	Algorithms	Role
Harvest HPUB	Retrieves a set of internal states based on per-chip identity	HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant)	User
Get Per-boot Nonce	Retrieves the unique per-boot nonce for Titan	HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant)	User
Alpha Unwrap Production Identity	Unwraps a Production Identity blob dataset	HKDF-SHA2-256 (non-conformant) EC DH P-256 (non-conformant) AES-CTR (non-conformant)	User
Alpha Production Identity Get Loaded Tokens	Retrieves set of BIOS tokens	HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant)	User
Alpha Signed Get Mode	Return modes support signed with Alpha Attestation Key	HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant)	User
Alpha Get Mode	Return modes supported	HKDF-SHA2-256 (non-conformant) ECDSA P-256 (non-conformant) AES-CTR (non-conformant)	User

Table 12: Non-Approved Services

4.5 External Software/Firmware Loaded

The module supports external firmware loaded for upgrades. An Approved ECDSA Signature Verification (P-256, SHA-256) firmware load test operation is performed prior to a firmware upgrade.

5 Software/Firmware Security

5.1 Integrity Techniques

The module bootloader and its operational firmware are verified separately using 32-bit EDC checksums. The checksum for the bootloader includes all the bootloader machine code. The checksum of the operational firmware image is verified by comparing a 32-bit checksum at run time with the checksum stored in the module which was computed at build time at the factory.

5.2 Initiate on Demand

The module integrity tests are 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 power cycling the module, which will perform (among others) the firmware integrity tests.

5.3 Additional Information

The temporary values generated during the module's integrity tests are zeroised upon completion of the integrity tests.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

The module is designed to accept only controlled firmware changes that successfully pass the software/firmware load test.

How Requirements are Satisfied:

The limited operational environment of the module prevents users from accessing SSPs which they are not authorized to access. There is no logical or physical access to the SSPs.

As per ISO/IEC 19790:2012 7.6.3:

- The cryptographic module has control over its own SSPs.
- The operational environment provides the capability to separate individual application processes from each other to prevent uncontrolled access to CSPs and uncontrolled modifications of SSPs, regardless if this data is in the process memory or stored on persistent storage within the operational environment.
- The module operates in a limited operational environment, therefore no restrictions or modifications to the configuration of the operational environment are possible.
- Processes that are spawned by the cryptographic module are owned by the module and are not owned by external processes/operators.

7 Physical Security

7.1 Mechanisms and Actions Required

The module is a single-chip cryptographic module made with production grade components, standard passivation techniques and standard IC packaging material.

Mechanism	Inspection Frequency	Inspection Guidance
N/A	N/A	N/A

Table 13: Mechanisms and Actions Required

8 Non-Invasive Security

Currently, the ISO/IEC 19790:2012 non-invasive security area is not required by FIPS 140-3 (see NIST SP 800-140F). The requirements of this area are not applicable to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The table below lists Sensitive Security Parameters (SSPs) storage areas for the module. Section 9.4 below selects from the storage areas listed and specifies the appropriate parameter in the "Storage" column if applicable to a specific SSP.

Storage Area Name	Description	Persistence Type
Flash Memory	Non-volatile	Static
RAM	Volatile memory	Dynamic

Table 14: Storage Areas

9.2 SSP Input-Output Methods

Per FIPS 140-3 IG 9.5.A Table 1 (CM Hardware to/from INT CM Hardware via INT Path (CM embedded within CM) this is N/A. No SSPs associated with the module exit or enter the module.

N/A for this module.

9.3 SSP Zeroization Methods

There are no unprotected SSP associated with the module. The table below lists Sensitive Security Parameters (SSPs) zeroization methods for the module. Section 9.4 below selects from the zeroization methods listed and specifies the appropriate parameter in the "Zeroization" column if applicable to a specific SSP.

Zeroization Method	Description	Rationale	Operator Initiation
Power Cycle	The contents of the module's volatile memory are zeroized on-demand by power cycling the module (removing power from the host device where the chip is inserted).	Volatile memory	Power Cycle

Table 15: SSP Zeroization Methods

9.4 SSPs

The following table summarizes the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module:

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
EEPROM Firmware Verification Key	Verify EEPROM firmware of the embedding River Redux module	256 - 128	Public Key - PSP	Factory Loaded		ECDSA SigVer (FIPS186-4) (A5419)
Payload Key	Validate the River Redux EEPROM firmware image and	2048 - 112	Public Key - PSP	Factory Loaded		RSA SigVer (FIPS186-4) (A5419)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	any firmware updates					
Downgrade Authorization Key	Authorise entering downgrade mode	256 - 128	Public Key - PSP	Factory Loaded		ECDSA SigVer (FIPS186-4) (A5419)

Table 16: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
EEPROM Firmware Verification Key		Flash Memory:Plaintext		N/A	
Payload Key		Flash Memory:Plaintext		N/A	
Downgrade Authorization Key		Flash Memory:Plaintext		N/A	

Table 17: SSP Table 2

9.5 Transitions

Per FIPS 140-3 IG C.K, FIPS 186-4 CAVP tests performed are mathematically identical to FIPS 186-5 CAVP tests, therefore the module can claim FIPS 186-5 compliance for these tests.

10 Self-Tests

This section specifies the pre-operational and conditional self-tests performed by the module. The pre-operational and conditional self-tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

10.1 Pre-Operational Self-Tests

Pre-operational Self-Tests are run upon the power up/initialization of the module. The module transitions to the operational state only after the pre-operational self-tests are passed successfully. The design of the module ensures that all data output, via the data output interface, is inhibited whenever the module is in a pre-operational self-test condition. The Pre-Operational Self-Tests are detailed in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
EDC Bootloader	32-bit	Error Detection Code	SW/FW Integrity	Status Output	Titan-BPN Chip Bootloader Integrity
EDC Firmware	32-bit	Error Detection Code	SW/FW Integrity	Status Output	Titan-BPN Chip Firmware Integrity

Table 18: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Conditional Self-Tests are run when an applicable security function or process is invoked. The Conditional Self-Tests are detailed in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256 (A5420)	256-bit hash	KAT	CAST	Successful initialization of the module	Hash	Module Initialization
SHA2-256 (A5419)	256-bit hash	KAT	CAST	Successful initialization of the module	Hash	Module Initialization
ECDSA SigVer (FIPS186-4) (A5419)	NIST P-256	KAT	CAST	Successful initialization of the module	Signature Verification	Module Initialization
RSA SigVer (FIPS186-4) (A5419)	2048-bit	KAT	CAST	Successful initialization of the module	Signature Verification	Module Initialization
Firmware Load Test	ECDSA SigVer (FIPS186-4) with NIST P-256	Load Test	SW/FW Load	Successful firmware update	Signature Verification	Firmware Update Request

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Repetition Count Test (RCT)	4096 raw 1 bit samples	FD	CAST	Successful output of entropy bits	NIST SP 800-90B Section 4.4.1	Continuous
Adaptative Proportion Test (APT)	4096 raw 1 bit samples	FD	CAST	Successful output of entropy bits	NIST SP 800-90B Section 4.4.2	Continuous

Table 19: Conditional Self-Tests

The module performs self-tests on all approved cryptographic algorithms supported in the approved mode of operation, using the tests shown in the table above. To ensure all conditional CASTs are performed prior to the first operational use of the associated algorithm, all CASTs are performed during the module's initial power-up sequence. Services are not available, and data output (via the data output interface) is inhibited during the applicable self-tests. If any of these tests fails, the module transitions to an error state.

10.3 Periodic Self-Test Information

Pre-operational self-tests can be run on-demand, for periodic testing, by power cycling the module.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
EDC Bootloader	Error Detection Code	SW/FW Integrity	On Demand	Power cycle
EDC Firmware	Error Detection Code	SW/FW Integrity	On Demand	Power cycle

Table 20: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256 (A5420)	KAT	CAST	On Demand	Power cycle
SHA2-256 (A5419)	KAT	CAST	On Demand	Power cycle
ECDSA SigVer (FIPS186-4) (A5419)	KAT	CAST	On Demand	Power cycle
RSA SigVer (FIPS186-4) (A5419)	KAT	CAST	On Demand	Power cycle
Firmware Load Test	Load Test	SW/FW Load	On Demand	Firmware Update Request
Repetition Count Test (RCT)	FD	CAST	On Demand	Entropy Bits Request
Adaptative Proportion Test (APT)	FD	CAST	On Demand	Entropy Bits Request

Table 21: Conditional Periodic Information

10.4 Error States

If any of the Pre-operational Self-Tests or Conditional Self-Tests fail, the module will output an error status and enter an error state, where all data output is inhibited. Upon entering an error state, an operator can attempt to clear the error state by power cycling the module. If the error state cannot be cleared, the module must be returned to the manufacturer.

The table below shows the different causes that lead to the Error States and the status indicators reported.

Name	Description	Conditions	Recovery Method	Indicator
Critical Error	The module's hard error state	POST or CAST Failure	Power Cycle	Error Code
Soft Error	The module's soft error state	Firmware load test failure	N/A	Error Code

Table 22: Error States

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Module is embedded in the River Redux Cryptographic Module (Network Interface Card).

No configuration of the module or installation steps are required from the operator. When the module is powered on its power-up self-tests are executed without any operator intervention. The module enters the Approved mode of operation automatically if the power-up self-tests complete successfully. If any of self-tests fail during power-up, the module will transition to an error state. The status of the module can be determined by the availability of the module. If the module is available, it has passed all self-tests. If it is unavailable, it is in the error state.

11.2 Administrator Guidance

None.

11.3 Non-Administrator Guidance

None.

12 Mitigation of Other Attacks

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