Cradlepoint, Inc.

Cradlepoint Kernel Cryptographic Module

Versions: 1.0, 2.0

FIPS 140-3 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.6

Prepared for:

Cradlepoint, Inc.
1100 W. Idaho Street, Floor 8
Boise, ID 83702-5389
United States of America

Phone: +1 888 331 2968
www.cradlepoint.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Abstract

This is a non-proprietary Cryptographic Module Security Policy for the Cradlepoint Kernel Cryptographic Module (versions 1.0 and 2.0) from Cradlepoint, Inc. (Cradlepoint). This Security Policy describes how the Cradlepoint Kernel Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-3, which details the U.S. and Canadian government requirements for cryptographic modules. More information about the FIPS 140-3 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in its Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-3 validation of the module. The Cradlepoint Kernel Cryptographic Module is referred to in this document as Cradlepoint Kernel Cryptographic Module or the module.

References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-3 cryptographic module security policy. More information is available on the module from the following sources:

- The Cradlepoint website www.cradlepoint.com contains information on the full line of services and solutions from Cradlepoint.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

Document Organization

ISO/IEC 19790 Annex B uses the same section naming convention as *ISO/IEC 19790* section 7 - Security requirements. For example, Annex B section B.2.1 is named “General” and B.2.2 is named “Cryptographic module specification,” which is the same as *ISO/IEC 19790* section 7.1 and section 7.2, respectively. Therefore, the format of this Security Policy is presented in the same order as indicated in Annex B, starting with “General” and ending with “Mitigation of other attacks.” If sections are not applicable, they have been marked as such in this document.

Table of Contents

1. General 5

2. Cryptographic Module Specification 7
2.1 Operational Environments 7
2.2 Algorithm Implementations 8
2.3 Cryptographic Boundary 10
2.4 Modes of Operation 11

3. Cryptographic Module Interfaces 12

4. Roles, Services, and Authentication 13
4.1 Authorized Roles 13
4.2 Authentication Methods 14
4.3 Services 14

5. Software/Firmware Security 16

6. Operational Environment 17

7. Physical Security 18

8. Non-Invasive Security 19

9. Sensitive Security Parameter Management 20
9.1 Keys and Other SSPs 20
9.2 DRBGs 21
9.3 SSP Storage Techniques 21
9.4 SSP Zeroization Methods 21
9.5 RGB Entropy Sources 21

10. Self-Tests 22
10.1 Pre-Operational Self-Tests 22
10.2 Conditional Self-Tests 22
10.3 On-Demand Self-Testing 23
10.4 Self-Test Failure Handling 23

11. Life-Cycle Assurance 24
11.1 Secure Installation 24
11.2 Initialization 24
11.3 Startup 24
11.4 Administrator Guidance 24
11.5 Non-Administrator Guidance 25

12. Mitigation of Other Attacks 26

Appendix A. Acronyms and Abbreviations 27

List of Tables

Table 1 – Security Levels....5
Table 2 – Tested Operational Environments....7
Table 3 – Vendor-Affirmed Operational Environments....7
Table 4 – Approved Algorithms....8
Table 5 – Approved Algorithms (provided by the Bound Module)....9
Table 6 – Non-Approved Algorithms Allowed in the Approved Mode of Operation....9
Table 7 – Ports and Interfaces....12
Table 8 – Roles, Service Commands, Input and Output....13
Table 9 – Approved Services....14
Table 10 – SSPs....20
Table 11 – Acronyms and Abbreviations....27

List of Figures

Figure 1 – Module Block Diagram....11

1. General

Cradlepoint is a global leader in cloud-delivered 4G and 5G wireless network edge solutions. Cradlepoint's NetCloud™ platform and cellular routers deliver a pervasive, secure, and software-defined Wireless WAN[1] edge to connect people, places, and things – anywhere. More than 28,500 businesses and government agencies worldwide, including many Global 2000 organizations and top public sector agencies, rely on Cradlepoint to keep mission-critical sites, points of commerce, field forces, vehicles, and IoT[2] devices always connected.

Cradlepoint NetCloud for Branch makes it easy to accelerate connecting to the Internet and critical applications from anywhere. Designed for traditional medium branches or locations requiring flexible connectivity, reliable performance, and simplified management, this all-in-one, compact endpoint includes full-featured routing, security, and Wi-Fi without needing extra hardware or complicated configurations.

The Cradlepoint Kernel Cryptographic Module is a cryptographic library running as part of the NetCloud operating system (OS) kernel that provides cryptographic services for Cradlepoint endpoints. The module offers symmetric encryption/decryption, digital signature verification, hashing, message authentication, and key establishment functions to support secure communications protocols.

The module uses the **Cradlepoint Cryptographic Module 1.0** (FIPS 140-3 certificate #4770) as a bound module to provide cryptographic support for the module's integrity testing. This requires an instance of the validated version of the Cradlepoint Cryptographic Module to be installed on the system for the primary module to operate in an Approved manner.

The Cradlepoint Kernel Cryptographic Module is validated at the FIPS 140-3 section levels shown in Table 1.

Table 1 – Security Levels

ISO/IEC 24579 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	N/A
8	Non-Invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-tests	1
11	Life-Cycle Assurance	1

[1] WAN – Wide Area Network
[2] IoT – Internet of Things

ISO/IEC 24579 Section 6. [Number Below]	FIPS 140-3 Section Title	Security Level
12	Mitigation of Other Attacks	N/A

The module has an overall security level of 1.

2. Cryptographic Module Specification

The Cradlepoint Kernel Cryptographic Module (versions 1.0 and 2.0) is a software module with a multi-chip standalone embodiment. The module is designed to operate within a modifiable operational environment.

The module comprises kernel loadable components, a static kernel binary, an integrity test utility, and digest files for testing integrity. All module components are contained within the host platform's physical enclosure.

2.1 Operational Environments

The module was tested and found to be compliant with FIPS 140-3 requirements on the environments listed in Table 2.

Table 2 – Tested Operational Environments

#	Operating System	Hardware Platform	Processor	PAA/Acceleration
1	NetCloud OS 7	Cradlepoint E3000	ARM Cortex-A (ARMv8-A)	Without

The vendor affirms the module's continued validation compliance when operating on the environments listed in Table 3.

Table 3 – Vendor-Affirmed Operational Environments

#	Operating System	Hardware Platform
1	NetCloud OS 7	Cradlepoint R920
2	NetCloud OS 7	Cradlepoint R2105/R2155
3	NetCloud OS 7	Cradlepoint R1900
4	NetCloud OS 7	Cradlepoint E300
5	NetCloud OS 7	Cradlepoint S700/S750
6	NetCloud OS 7	Cradlepoint R980
7	NetCloud OS 7	Cradlepoint E3000

The cryptographic module maintains validation compliance when operating on any general-purpose computer (GPC) provided that the GPC uses any single-user operating system/mode specified on the validation certificate, or another compatible single-user operating system. The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment not listed on the validation certificate.

The sections below describe the module boundary, algorithm implementations, and modes of operation.

2.2 Algorithm Implementations

Validation certificates for each Approved security function are listed in Table 4 below. Note that there are algorithms, modes, and key/moduli sizes that have been CAVP-tested but are not used by any Approved service of the module. The update from version 1.0 to version 2.0 removes all Triple DES functions and adds AES-CTR encryption/decryption as an Approved algorithm. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in Table 4 are used by an Approved service of the module.

Table 4 – Approved Algorithms

CAVP Certificate	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths	Use / Function
A3232 A4944	**AES**[3] *FIPS PUB[4] 197* *NIST SP 800-38A*	CBC[5], ECB[6]	128, 192, 256	Encryption/decryption
A4944	**AES** *FIPS PUB[7] 197* *NIST SP 800-38A*	CTR	128, 192, 256	Encryption/decryption *ACS-CTR functionality is available only in version 2.0 of the module.*
A3232 A4944	**AES** *FIPS PUB[8] 197* *NIST SP 800-38D*	GCM[9] (external IV)	128, 192, 256	Encryption/decryption
A3232 A4944	**HMAC**[10] *FIPS PUB 198-1*	SHA-1, SHA2-256, SHA2-384, SHA2-512	112 (minimum)	Message authentication
A3232 A4944	**KTS**[11] *NIST SP 800-38D*	AES-GCM	128, 192, 256	Key wrap/unwrap (authenticated encryption/decryption)[12] *SSP establishment methodology provides between 128 and 256 bits of encryption strength.*
A3232 A4944	**SHS**[13] *FIPS PUB 180-4*	SHA-1, SHA2-256, SHA2-384, SHA2-512	-	Message digest

[3] AES – Advanced Encryption Standard
[4] PUB – Publication
[5] CBC – Cipher Block Chaining
[6] ECB – Electronic Codebook
[7] PUB – Publication
[8] PUB – Publication
[9] GCM – Galois/Counter Mode
[10] HMAC – (Keyed-) Hash Message Authentication Code
[11] KTS – Key Transport Scheme
[12] Per *FIPS 140-3 Implementation Guidance* D.G, AES-GCM is an Approved key transport technique.
[13] SHS – Secure Hash Standard

CAVP Certificate	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths	Use / Function
A3232	**Triple-DES**[14] *NIST SP 800-67rev2* *NIST SP 800-38A*	CBC	168	Decryption *Triple-DES functionality is available only in version 1.0 of the module.*

Table 5 lists the Approved algorithms that are provided by the bound module and used by this module in the Approved mode. The table includes only those algorithms used by the primary module.

Table 5 – Approved Algorithms (provided by the Bound Module)

CAVP Certificate	Algorithm and Standard	Mode / Method	Description/ Key Size(s) / Key Strength(s)	Use / Function
A2584	**HMAC** *FIPS PUB 198-1*	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	112 (minimum)	Message authentication
A2584	**SHS** *FIPS PUB 180-4*	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	-	Message digest

The module implements the non-Approved but allowed algorithms shown in Table 6 below.

Table 6 – Non-Approved Algorithms Allowed in the Approved Mode of Operation

Algorithm	Caveat	Use / Function
AES (Cert. A3232, A4944)	SSP establishment methodology provides between 128 and 256 bits of encryption strength.	Key unwrapping (using any Approved mode) [15]
Triple-DES (Cert. A3232)	SSP establishment methodology provides 168 bits of encryption strength.	Key unwrapping (using any Approved mode with two-key or three-key) [16] *Triple-DES functionality is available only in version 1.0 of the module.*

The module does not implement any non-Approved algorithms allowed in the Approved mode of operation for which no security is claimed.

The module does not implement any non-Approved algorithms not allowed in the Approved mode of operation.

[14] DES – Data Encryption Standard
[15] Per FIPS 140-3 Implementation Guidance D.G, key unwrapping using any Approved mode of AES is an allowed method for key transport.
[16] Per FIPS 140-3 Implementation Guidance D.G, key unwrapping using any Approved mode of two-key or three-key Triple-DES is an allowed method for key transport.

2.3 Cryptographic Boundary

As a software cryptographic module, the module has no physical components. The physical perimeter of the cryptographic module is defined by the host platform on which the module is installed. The module's cryptographic boundary comprises all functionalities contained within the module's compiled source code. Several of the module's components are contained within a FIT[17] image called fit.itb.padded. These components are:

- Version 1.0
  - /lib/modules/4.4.100-coconut/kernel/crypto/*.ko (cryptographic kernel object files)
  - /lib/modules/4.4.100-coconut/kernel/crypto/ocf/*.ko (cryptographic kernel object files)

- Version 2.0
  - /lib/modules/5.4.164-coconut/kernel/crypto/*.ko (cryptographic kernel object files)
  - /lib/modules/5.4.164-coconut/kernel/crypto/ocf/*.ko (cryptographic kernel object files)

The remaining components are:

- /dev/mmcblk0pX (static kernel binary partition)
- /service_manager/verify_kernel_hmac.pyc (integrity test utility)
- /etc/rc (script file for executing integrity test utility)
- /etc/hmacs (file containing the HMAC digest values for the FIT image, integrity test utility, and rc script)

The cryptographic boundary is the contiguous perimeter that surrounds all memory-mapped functionality provided by the module when loaded and stored in the host platform's memory. The module is entirely contained within the physical perimeter of the host platform.

Figure 1 shows the logical block diagram of the module executing in memory, its location with respect to the operating system and other supporting applications, and its interactions with surrounding software components, as well as the host platform's physical perimeter and module's cryptographic boundary.

[17] FIT – Flattened uImage Tree

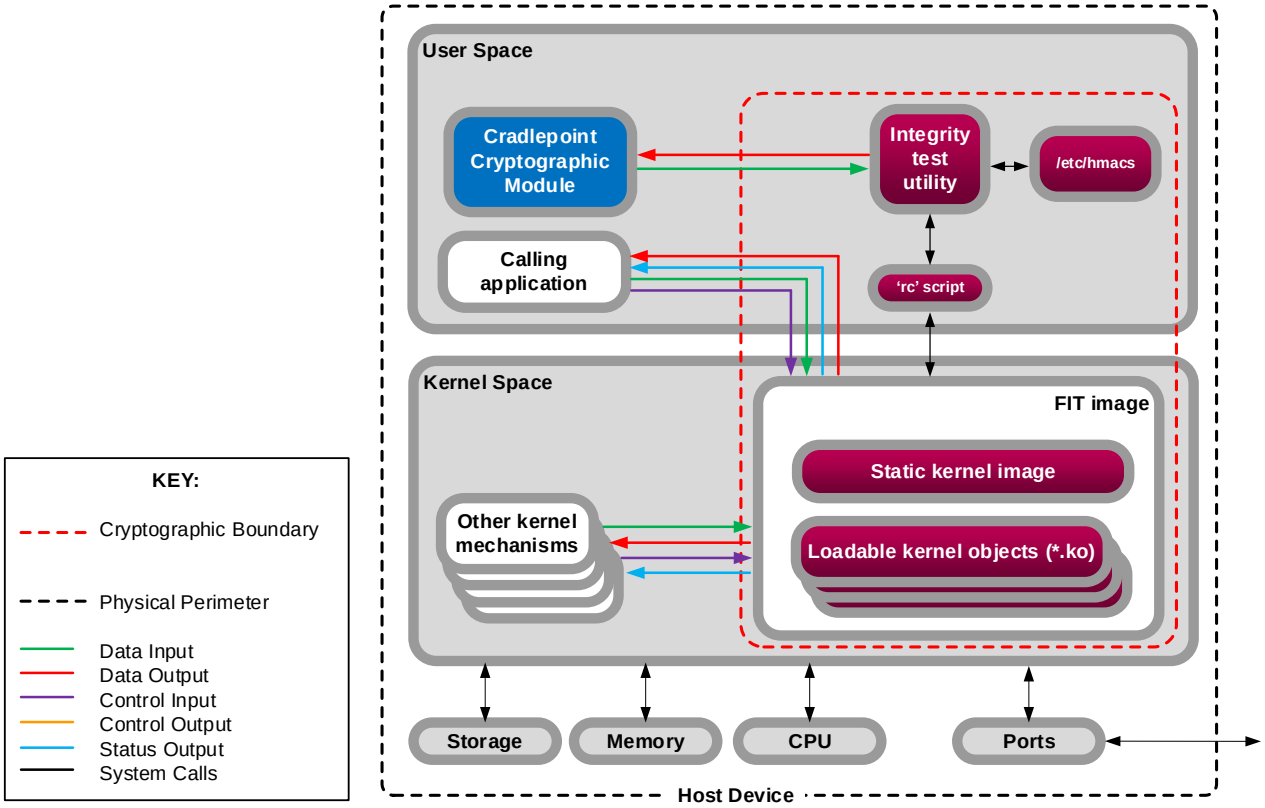

Figure 1 – Module Block Diagram

2.4 Modes of Operation

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

3. Cryptographic Module Interfaces

FIPS 140-3 defines the following logical interfaces for cryptographic modules:

- Data Input
- Data Output
- Control Input
- Control Output
- Status Output

As a software library, the cryptographic module has no direct access to any of the host platform's physical ports/interfaces. The logical interfaces are the kernel-level APIs[18] by which module services are requested. A mapping of the FIPS-defined interfaces and the module's ports and interfaces can be found in Table 7. Note that the module does not output control information, and thus has no logical interface specified for control output.

Table 7 – Ports and Interfaces

Physical Port	Logical Interface	Data That Passes Over Port/Interface
Physical data input port(s) of the tested platforms	Data Input • API input arguments that provide input data for processing	• Data to be encrypted, decrypted, signed, verified, or hashed • Keys to be used in cryptographic services
Physical data output port(s) of the tested platforms	Data Output • API output arguments that return generated or processed data back to the caller	• Data that has been encrypted, decrypted, or verified • Digital signatures • Hashes • Keys established using module's key establishment methods
Physical control input port(s) of the tested platforms	Control Input • API input arguments that are used to initialize and control the operation of the module	• API commands invoking cryptographic services • Modes, key sizes, etc. used with cryptographic services
Physical status output port(s) of the tested platforms	Status Output • API call return values	• Status information regarding the module • Status information regarding the invoked service/operation

[18] API – Application Programming Interface

4. Roles, Services, and Authentication

The sections below describe the module's authorized roles, services, and operator authentication methods.

4.1 Authorized Roles

The module supports the following role(s) that authorized operators can assume:

- Crypto Officer (CO) – The CO role performs cryptographic initialization or management functions and general security services.
- User – The User role performs general security services, including cryptographic operations and other approved security functions.

The module does not support multiple concurrent operators. The calling application that loaded the module is its only operator.

Table 8 below lists the supported roles, along with the services (including input and output) available to each role.

Table 8 – Roles, Service Commands, Input and Output

Role	Service	Input	Output
CO	Show Status	API call	Current operational status
CO	Perform self-tests on-demand	Reboot or power-cycle host platform	Status
CO	Zeroize	API call	Status
CO	Show versioning information	API call parameters	Module name, version
User	Perform symmetric encryption	API call parameters, key, plaintext	Status, ciphertext
User	Perform symmetric decryption	API call parameters, key, ciphertext	Status, plaintext
User	Perform authenticated symmetric encryption	API call parameters, key, plaintext	Status, ciphertext
User	Perform authenticated symmetric decryption	API call parameters, key, ciphertext	Status, plaintext
User	Perform keyed hash operations	API call parameters, key, message	Status, MAC[19]
User	Perform hash operation	API call parameters, message	Status, hash
User	Perform key wrap	API call parameters, wrapping key, key	Status, hash
User	Perform key unwrap	API call parameters, wrapping key, key	Status, hash

[19] MAC – Message Authentication Code

4.2 Authentication Methods

The module does not support authentication methods; operators implicitly assume an authorized role based on the service selected.

4.3 Services

Descriptions of the services available to the authorized roles are provided in Table 9 below.

The module is an integrated component of Cradlepoint's NetCloud OS and offers crypto functions to applications and other kernel components installed on Cradlepoint devices. While the module includes implementations of non-Approved security functions that can be called by other Cradlepoint kernel modules, all such invocations will return failure codes to the caller. This effectively limits the service offerings to Approved services only.

As allowed for this scenario per section 2.4.C of *FIPS 140-3 Implementation Guidance*, the module provides indicators for the use of Approved services through a combination of an explicit indication (via a Global Approved mode indicator) and an implicit indication (via the API return value of the service).

The keys and Sensitive Security Parameters (SSPs) listed in the table indicate the type of access required using the following notation:

- 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 9 – Approved Services

Service	Description	Approved Security Function(s)	Keys and/or SSPs	Role	Access Rights to Keys and/or SSPs	Indicator
Show status	Return mode status	None	None	CO	None	N/A
Zeroize	Zeroize and de-allocate memory containing sensitive data	None	All SSPs	CO	All SSPs – Z	N/A
Show versioning information	Return module versioning information	None	None	CO	None	N/A
Perform symmetric encryption	Encrypt plaintext data	AES (CBC, CTR, ECB modes, all supported key sizes) (Certs. A3232, A4944)	AES key	User	AES key – WE	Global Approved mode indicator
Perform symmetric decryption	Decrypt ciphertext data	AES (CBC, CTR, ECB modes, all supported key sizes) (Certs. A3232, A4944) Triple-DES (CBC mode, all supported key sizes) (Cert. A3232)	AES key Triple-DES key	User	AES key – WE Triple-DES key – WE	Global Approved mode indicator
Perform authenticated symmetric encryption	Encrypt plaintext using supplied AES GCM key	AES (GCM mode, all supported key sizes) (Certs. A3232, A4944)	AES GCM key AES GCM IV	User	AES GCM key – WE AES GCM IV – WE	Global Approved mode indicator

Service	Description	Approved Security Function(s)	Keys and/or SSPs	Role	Access Rights to Keys and/or SSPs	Indicator
Perform authenticated symmetric decryption	Decrypt ciphertext using supplied AES GCM key and external IV[20]	AES (GCM mode, all supported key sizes) (Certs. A3232, A4944)	AES GCM key AES GCM IV	User	AES GCM key – WE AES GCM IV – WE	Global Approved mode indicator
Perform keyed hash operations	Compute a message authentication code	HMAC (all supported SHAs) (Certs. A3232, A4944) SHS (all supported hash sizes) (Certs. A3232, A4944)	HMAC key	User	HMAC key – WE	Global Approved mode indicator
Perform hash operation	Compute a message digest	SHS (all supported hash sizes) (Certs. A3232, A4944)	None	User	None	Global Approved mode indicator
Perform key wrap	Perform key wrap	AES (GCM mode, all supported key sizes) (Certs. A3232, A4944)	AES GCM key AES GCM IV	User	AES GCM – WE AES GCM IV – WE	Global Approved mode indicator
Perform key unwrap	Perform key unwrap	AES (CBC, CTR, ECB modes, all supported key sizes) (Certs. A3232, A4944) AES (GCM mode, all supported key sizes) (Certs. A3232, A4944) Triple-DES (CBC mode, all supported key sizes) (Cert. A3232)	AES key AES GCM key AES GCM IV Triple-DES key	User	AES key – WE AES GCM – WE AES GCM IV – WE Triple-DES key – WE	Global Approved mode indicator

Per FIPS 140-3 Implementation Guidance 2.4.C, the **Show Status, **Zeroize**, and **Show Versioning Information** services do not require an Approved security service indicator.*

***Triple-DES functionality is available only in version 1.0 of the module.*

The module does not support a non-Approved mode of operation and offers no non-Approved services.

[20] IV – Initialization Vector

5. Software/Firmware Security

All software components within the cryptographic boundary are verified when they are loaded into memory during boot time using approved integrity techniques (refer to section 10.1 for details regarding the module's integrity test techniques). The module implements an HMAC SHA2-256 for the integrity test of each module component; failure of a test will cause the kernel to panic, and the module will enter a critical error state.

The module's integrity check is performed automatically at module instantiation (i.e., when the module is loaded into memory for execution) without action from the module operator. The CO can initiate the pre-operational tests on demand by power-cycling the host platform or rebooting the OS.

The Cradlepoint Kernel Cryptographic Module is not delivered to end-users as a standalone offering. Rather, it is a pre-built integrated component of Cradlepoint's solutions. Cradlepoint does not provide end-users with any mechanisms to directly access the module, its source code, its APIs, or any information sent to/from the module. Thus, end-users have no ability to independently load the module onto target platforms. No configuration steps are required to be performed by end-users, and no end-user action is required to initialize the module for operation.

6. Operational Environment

The Cradlepoint Kernel Cryptographic Module comprises a software cryptographic library that executes in a modifiable operational environment.

The cryptographic module has control over its own SSPs. The process and memory management functionality of the host platform's OS prevents unauthorized access to plaintext private and secret keys, intermediate key generation values and other SSPs by external processes during module execution. The module only allows access to SSPs through its well-defined API. The operational environment provides the capability to separate individual application processes from each other by preventing uncontrolled access to CSPs and uncontrolled modifications of SSPs regardless of whether this data is in the process memory or stored on persistent storage within the operational environment. Processes that are spawned by the module are owned by the module and are not owned by external processes/operators.

Please refer to section 2.1 of this document for a list/description of the applicable operational environments.

7. Physical Security

The cryptographic module is software module and does not include physical security mechanisms. Therefore, per *ISO/IEC 19790:2021* section 7.7.1, requirements for physical security are not applicable.

8. Non-Invasive Security

This section is not applicable. There are currently no approved non-invasive mitigation techniques referenced in *ISO/IEC 19790:2021* Annex F.

9. Sensitive Security Parameter Management

9.1 Keys and Other SSPs

The module supports the keys and other SSPs listed Table 10.

Table 10 – SSPs

Key/SSP Name/Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establishment	Storage	Zeroization	Use & Related Keys
Keys								
AES key (CSP)	Between 128 and 256 bits	AES (CBC, ECB modes) (Certs. A3232, A4944) AES (CTR mode) (Certs. A4944) KTS (Certs. A3232, A4944)	-	Imported in plaintext via API parameter Never exported	-	Not persistently stored by the module	Via zeroization command; reboot/power-cycle the host device	Symmetric encryption and decryption, key unwrap
AES GCM key (CSP)	Between 128 and 256 bits	AES (GCM mode) (Certs. A3232, A4944) KTS (Certs. A3232, A4944)	-	Imported in plaintext via API parameter Never exported	-	Not persistently stored by the module	Via zeroization command; reboot/power-cycle the host device	Authenticated symmetric encryption and decryption, key wrap and unwrap
Triple-DES key (CSP)	-	Triple-DES (Cert. A3232) KTS (Cert. A3232)	-	Imported in plaintext via API parameter Never exported	-	Not persistently stored by the module	Via zeroization command; reboot/power-cycle the host device	Symmetric decryption, key unwrap *Triple-DES functionality is available only in version 1.0 of the module.*
HMAC key (CSP)	160-bit (minimum) key	HMAC (Certs. A3232, A4944)	-	Imported in plaintext via API parameter Never exported	-	Not persistently stored by the module	Via zeroization command; reboot/power-cycle the host device	Keyed hash
Other SSPs								
AES GCM IV (CSP)	-	AES (GCM mode) (Certs. A3232, A4944) KTS (Certs. A3232, A4944)	-	Imported in plaintext via API parameter Never exported		Not persistently stored by the module	Via zeroization command; reboot/power-cycle the host device	Initialization vector for AES GCM

The calling application (running outside the module's cryptographic boundary) supports IPsec-v3. This application negotiates the protocol session's keys and the value in the first 32 bits of the nonce, with the last 64 bits being deterministic. It relies on the cryptographic module to perform the AES GCM encryption for ESP within IPsec-v3. The AES GCM IV generation method complies with scenario #1 in *FIPS 140-3 IG C.H*.

The GCM IV construction is compliant with *RFC 4106* and *RFC 5282*. The IV is only used in the context of the AES GCM mode encryption within the IPsec-v3 protocol. When the IV exhausts the maximum number of possible values for a given security association, either party to the security association that encounters this condition triggers a rekeying with IKEv2 to establish a new encryption key for the security association.

9.2 DRBGs

The module includes no DRBGs; it does not generate cryptographic keys or seeds for the generation of cryptographic keys.

9.3 SSP Storage Techniques

There is no mechanism within the module's cryptographic boundary for the persistent storage of SSPs. The module uses SSPs passed in on the stack by the calling application and does not store these SSPs beyond the lifetime of the API call.

9.4 SSP Zeroization Methods

As a software cryptographic module, there is no mechanism within the module boundary for the persistent storage of keys and CSPs. Maintenance, including protection and zeroization, of any keys and CSPs that exist outside the module's cryptographic boundary is the responsibility of the end-user.

The application that uses the module is responsible for appropriate destruction and zeroization of the key material. The module provides functions for key allocation and destruction, which overwrites the memory that is occupied by the key information with zeroes before it is deallocated. When a calling application calls the appropriate API function, that operation overwrites memory with zeroes and then frees that memory.

Memory is automatically overwritten by zeroes when freeing the cipher handler. The following methods are available:

- `crypto_free_cipher()` – key zeroization for single raw cipher handle
- `crypto_free_ablkcipher()` – key zeroization method for asynchronous raw cipher handle
- `crypto_free_blkcipher ()` – key zeroization method for synchronous raw cipher handle
- `crypto_free_skcipher()` – key zeroization method for symmetric ciphers
- `crypto_free_shash()` – key zeroization method for synchronous message digest handle

9.5 RGB Entropy Sources

The module requires no entropy sources, as it does not implement an Approved DRBG.

10. Self-Tests

Both pre-operational and conditional self-tests are performed by the module. Pre-operational tests are performed between the time the cryptographic module is instantiated and before the module transitions to the operational state. Conditional self-tests are performed by the module during module operation when certain conditions exist. The following sections list the self-tests performed by the module, their expected error status, and the error resolutions.

10.1 Pre-Operational Self-Tests

The module performs the following pre-operational self-tests:

- Software integrity test for the integrity test utility using an HMAC SHA2-256 digest
- Software integrity test for the FIT image file using an HMAC SHA2-256 digest
- Software integrity test for the 'rc' script file using an HMAC SHA2-256 digest

When the kernel boots, the module begins to load. Upon loading, the module automatically executes the module's cryptographic algorithm self-tests (CASTs). If all CASTs are successful, the module then executes the pre-operational integrity tests by calling an 'rc' script that invokes the integrity test utility. This utility leverages the HMAC and SHA2-256 implementations from the bound module to compute one digest from the FIT file image, a second digest from its own image, and a third digest for the 'rc' script. The integrity test utility then compares the newly-computed digest values to the pre-computed digest values in order to make the determination of pass or fail.

The HMAC and SHA2-256 implementations used by the module's integrity test utility to verify to the module's HMAC digest files are provided by the bound module operating in its Approved mode of operation. As required by FIPS 140-3, the HMAC and SHA KATs from the bound module are performed when the bound module is loaded for execution and prior to the primary module's invocation of these algorithms for the integrity tests.

10.2 Conditional Self-Tests

The module performs the following conditional self-tests:

- Conditional cryptographic algorithm self-tests (CASTs)
  - AES CBC encrypt KAT[21] (128-bit)
  - AES CBC decrypt KAT (128-bit)
  - AES ECB encrypt KAT (128-bit)
  - AES ECB decrypt KAT (128-bit)
  - AES GCM encrypt KAT (128-bit)
  - AES GCM decrypt KAT (128-bit)
  - HMAC KATs (SHA-1, SHA2-256, SHA2-384, SHA2-512)
  - SHA-1 KAT
  - SHA-2 KATs (SHA2-256, SHA2-384, SHA2-512)

[21] KAT – Known Answer Test

- Triple-DES CBC decrypt KAT (3-Key)[22]

10.3 On-Demand Self-Testing

Module operators can initiate the pre-operational self-tests and conditional CASTs on demand and for periodic testing of the module by power-cycling the host platform or rebooting the OS. This will again call the 'rc' script to perform the module's CASTs and pre-operational integrity test using the HMAC and SHA implementations provided by the bound module.

10.4 Self-Test Failure Handling

Upon failure of any self-test, the module logs an error message to the kernel ring buffer and enters a critical error state. In this state, the kernel is panicked, and the module will not load; hence, the module will have no ability to perform cryptographic services or output data over the data output interfaces. Evidence of the failure can be observed by module operators by entering the '`log`' command at the command line.

To recover, the host platform must be rebooted or power-cycled. If the pre-operational self-tests complete successfully, then the module can resume normal operations. If the module continues to experience self-test failures after reinitializing, then the module will not be able to resume normal operations, and the CO should contact Cradlepoint, Inc. for assistance.

[22] Triple-DES self-tests are performed only in version 1.0 of the module.

11. Life-Cycle Assurance

The sections below describe how to ensure the module is operating in its validated configuration, including the following:

- Procedures for secure installation, initialization, startup, and operation of the module
- Maintenance requirements
- Administrator and non-Administrator guidance

Operating the module without following the guidance herein (including the use of undocumented services) will result in non-compliant behavior and is outside the scope of this Security Policy.

11.1 Secure Installation

The module is an integrated component of Cradlepoint's proprietary NetCloud operating system and is pre-installed on Cradlepoint devices prior to distribution to end-users; thus, no independent installation steps are required.

Cradlepoint does not provide any mechanisms for end-users to directly access the module, its source code, its APIs, or any information sent between the module and the calling application.

11.2 Initialization

This module is designed to support Cradlepoint devices; Cradlepoint's applications and kernel modules are the sole consumers of the cryptographic services provided by the module. No end-user action is required to initialize the module for operation; the calling application performs any actions required to initialize the module.

The pre-operational integrity test and cryptographic algorithm self-tests are performed automatically when the module is loaded for execution, without any specific action from the calling application or the end-user. End-users have no means to short-circuit or bypass these actions. Failure of any of the initialization actions will result in a failure of the module to load for execution.

11.3 Startup

No setup steps are required to be performed by end-users.

11.4 Administrator Guidance

There are no specific management activities required of the CO role to ensure that the module runs securely. However, if any irregular activity is noticed or the module is consistently reporting errors, then Cradlepoint Customer Support should be contacted.

The following list provides additional guidance for module administrators:

- The module outputs a mode indicator upon completion of the pre-operational self-test and CASTs. The CO can review this value to determine the module's operational status. When the module is in the Approved mode, the CO will find the value "1" written in the file `/proc/sys/crypto/fips_enabled`. Otherwise, the file will contain "0" (or will not exist at all).

- The `crypto_get_version()` API command can be used to obtain the module's versioning information. This information will include the module name and version, which can be correlated with the module's validation record.

11.5 Non-Administrator Guidance

The following list provides additional policies for non-administrators:

- In the event that power to the module is lost and subsequently restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are re-distributed.

12. Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-3 Level 1 requirements for this validation.

Appendix A. Acronyms and Abbreviations

Table 11 provides definitions for the acronyms and abbreviations used in this document.

Table 11 – Acronyms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CO	Cryptographic Officer
CSP	Critical Security Parameter
CTR	Counter
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
ECB	Electronic Codebook
FIPS	Federal Information Processing Standard
FIT	Flattened uImage Tree
GCM	Galois/Counter Mode
GPC	General-Purpose Computer
HMAC	(Keyed-) Hash Message Authentication Code
IKE	Internet Key Exchange
IoT	Internet of Things
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
IV	Initialization Vector
KAT	Known Answer Test
KTS	Key Transport Scheme
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OS	Operating System
PSP	Public Security Parameter
PUB	Publication
SHA	Secure Hash Algorithm

Term	Definition
SHS	Secure Hash Standard
SP	Special Publication
SSP	Sensitive Security Parameter
WAN	Wide Area Network

Prepared by:
Corsec Security, Inc.

12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com