

# L3Harris Technologies, Inc.

## Harris AES Load Module

Version: R07A01

### FIPS 140-3 Non-Proprietary Security Policy

FIPS Security Level: 1

Document Version: 0.1

Prepared for:



**L3Harris Technologies**  
221 Jefferson Ridge Parkway  
Lynchburg, VA 24501  
United States of America

Phone: +1 434 316-7181  
<http://www.l3harris.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](http://www.corsec.com)

# Table of Contents

---

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>1. General.....</b>                                                 | <b>5</b>  |
| 1.1 Overview .....                                                     | 5         |
| 1.2 Security Levels.....                                               | 5         |
| <b>2. Cryptographic Module Specification .....</b>                     | <b>7</b>  |
| 2.1 Description.....                                                   | 7         |
| 2.2 Tested and Vendor Affirmed Module Version and Identification ..... | 9         |
| 2.3 Excluded Components .....                                          | 10        |
| 2.4 Modes of Operation.....                                            | 10        |
| 2.5 Algorithms.....                                                    | 11        |
| 2.6 Security Function Implementations.....                             | 12        |
| 2.7 Algorithm Specific Information .....                               | 13        |
| 2.8 RBG and Entropy .....                                              | 13        |
| 2.9 Key Generation .....                                               | 13        |
| 2.10 Key Establishment.....                                            | 13        |
| 2.11 Industry Protocols.....                                           | 13        |
| <b>3. Cryptographic Module Interfaces .....</b>                        | <b>14</b> |
| 3.1 Ports and Interfaces.....                                          | 14        |
| <b>4. Roles, Services, and Authentication .....</b>                    | <b>15</b> |
| 4.1 Authentication Methods.....                                        | 15        |
| 4.2 Roles.....                                                         | 15        |
| 4.3 Approved Services .....                                            | 15        |
| 4.4 Non-Approved Services .....                                        | 18        |
| 4.5 External Software/Firmware Loaded .....                            | 18        |
| 4.6 Additional Information .....                                       | 19        |
| <b>5. Software/Firmware Security .....</b>                             | <b>20</b> |
| 5.1 Integrity Techniques .....                                         | 20        |
| 5.2 Initiate on Demand .....                                           | 20        |
| <b>6. Operational Environment.....</b>                                 | <b>21</b> |
| 6.1 Operational Environment Type and Requirements .....                | 21        |
| <b>7. Physical Security .....</b>                                      | <b>22</b> |
| <b>8. Non-Invasive Security .....</b>                                  | <b>23</b> |
| <b>9. Sensitive Security Parameters Management .....</b>               | <b>24</b> |
| 9.1 Storage Areas.....                                                 | 24        |
| 9.2 SSP Input-Output Methods.....                                      | 24        |
| 9.3 SSP Zeroization Methods .....                                      | 24        |
| 9.4 SSPs .....                                                         | 24        |
| 9.5 Additional Information .....                                       | 25        |
| <b>10. Self-Tests.....</b>                                             | <b>26</b> |
| 10.1 Pre-Operational Self-Tests .....                                  | 26        |

|            |                                                            |           |
|------------|------------------------------------------------------------|-----------|
| 10.2       | Conditional Self-Tests .....                               | 26        |
| 10.3       | Periodic Self-Test Information .....                       | 27        |
| 10.4       | Error States .....                                         | 27        |
| 10.5       | Operator Initiation of Self-Tests .....                    | 27        |
| <b>11.</b> | <b>Life-Cycle Assurance.....</b>                           | <b>28</b> |
| 11.1       | Installation, Initialization, and Startup Procedures ..... | 28        |
| 11.2       | Administrator Guidance.....                                | 28        |
| 11.3       | Non-Administrator Guidance.....                            | 29        |
| 11.4       | Design and Rules.....                                      | 29        |
| 11.5       | End of Life .....                                          | 29        |
| <b>12.</b> | <b>Mitigation of Other Attacks.....</b>                    | <b>30</b> |
|            | <b>Appendix A. Acronyms and Abbreviations .....</b>        | <b>31</b> |

# List of Tables

---

Table 1: Security Levels .....6

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) .....9

Table 3: Tested Operational Environments - Software, Firmware, Hybrid ..... 10

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid..... 10

Table 5: Modes List and Description ..... 11

Table 6: Approved Algorithms..... 11

Table 7: Security Function Implementations..... 12

Table 8: Ports and Interfaces..... 14

Table 9: Roles..... 15

Table 10: Approved Services ..... 18

Table 11: Storage Areas..... 24

Table 12: SSP Input-Output Methods..... 24

Table 13: SSP Zeroization Methods..... 24

Table 14: SSP Table 1..... 25

Table 15: SSP Table 2..... 25

Table 16: Pre-Operational Self-Tests..... 26

Table 17: Conditional Self-Tests ..... 26

Table 18: Pre-Operational Periodic Information ..... 27

Table 19: Conditional Periodic Information ..... 27

Table 20: Error States ..... 27

Table 21. Acronyms and Abbreviations..... 31

# List of Figures

---

Figure 1. Module Block Diagram .....8

Figure 2. Radio Terminal Block Diagram.....9

# 1. General

---

## 1.1 Overview

This is a non-proprietary Cryptographic Module Security Policy for the Harris AES Load Module (version: R07A01) from L3Harris Technologies (L3Harris). This Security Policy describes how the Harris AES Load 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 a secure Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-3 validation of the module. The Harris AES Load Module is referred to in this document as HALM or the module.

### 1.1.1 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 L3Harris website <https://www.l3harris.com/> contains information on the full line of services and solutions from L3Harris.
- The validated modules 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.

### 1.1.2 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.

## 1.2 Security Levels

The Harris AES Load Module is validated at the FIPS 140-3 section levels shown in the table below.

| Section | Title                              | Security Level |
|---------|------------------------------------|----------------|
| 1       | General                            | 1              |
| 2       | Cryptographic module specification | 1              |
| 3       | Cryptographic module interfaces    | 1              |

| Section | Title                                   | Security Level |
|---------|-----------------------------------------|----------------|
| 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**

## 2. Cryptographic Module Specification

---

### 2.1 Description

#### 2.1.1 Purpose and Use

L3Harris Technologies is a leading supplier of systems and equipment for public safety, federal, utility, commercial, and transportation markets. Their products range from the most advanced IP<sup>1</sup> voice and data networks, to industry-leading multiband/multimode radios, and even public safety-grade broadband video and data solutions. Their comprehensive line of software-defined radio products and systems support the critical missions of countless public and private agencies, federal and state agencies, and government, defense, and peacekeeping organizations throughout the world.

The Harris AES Load Module supports secure voice and data communications by providing AES <sup>2</sup> encryption/decryption as specified in *FIPS PUB 197*. The module ensures data integrity using a Cipher-based Message Authentication Code (CMAC) algorithm as specified in *NIST SP 800-38B*. The module interacts with a Digital Signal Processor (DSP) application executing on the L3Harris XL family of radios and other terminal products in order to provide its services to those terminals. The Module Software Interface (MSI) is the mechanism by which the DSP can (a) request the HALM to perform specific processing, (b) provide the HALM with specific parameter values, or (c) request a particular status response from the HALM.

The application code running on the host radio terminal sends any requests for specific HALM processing, status, or parameter updates to the DSP application. The DSP application then communicates these requests to the HALM via the MSI. If any information needs to be sent from the HALM to the host, the HALM returns an appropriate status value to the DSP Application along with any relevant data in the Data Output port of the interface. The DSP Application then forms the appropriate message and sends the message to the host.

#### 2.1.2 Module Type

The Harris AES Load Module R07A01 is a Firmware module.

#### 2.1.3 Module Embodiment

The Harris AES Load Module has a Multi-Chip Standalone embodiment.

#### 2.1.4 Cryptographic Boundary

Figure 1 below is a logical block diagram showing the module's cryptographic boundary and physical perimeter, as well as its interactions with surrounding firmware components as it executes in the DSP's SRAM.

---

<sup>1</sup> IP – Internet Protocol

<sup>2</sup> AES – Advanced Encryption Standard

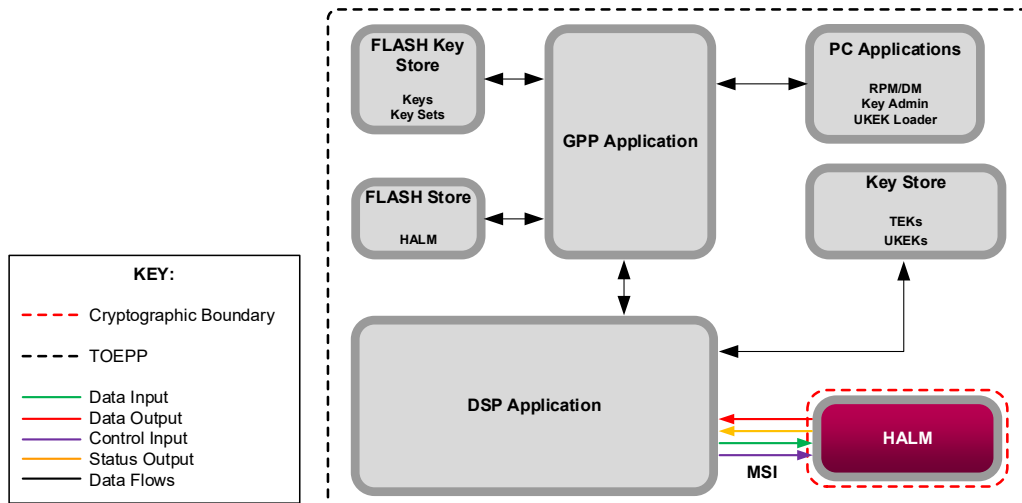


**Figure 1. Module Block Diagram**

The L3Harris terminals do not employ an operating system (OS). Rather, L3Harris terminals rely on the L3Harris BIOS<sup>3</sup> Kernel version 1.0 to act as the module's "operating system", providing the low-level interface to the underlying hardware. The kernel executes on an internal Blackfin BF707 DSP<sup>4</sup> (with a Blackfin+ core embedded processor) from Analog Devices, Inc. The module provides cryptographic services directly to the DSP application.

## 2.1.5 Tested Operational Environment's Physical Perimeter (TOEPP)

Figure 2 below is a block diagram of the major hardware components in the L3Harris terminals that host the HALM (including an NXP/Freescale processor), along with the terminal's enclosure which comprises the TOEPP.

<sup>3</sup> BIOS – Basic Input/Output System

<sup>4</sup> DSP - Digital Signal Processor

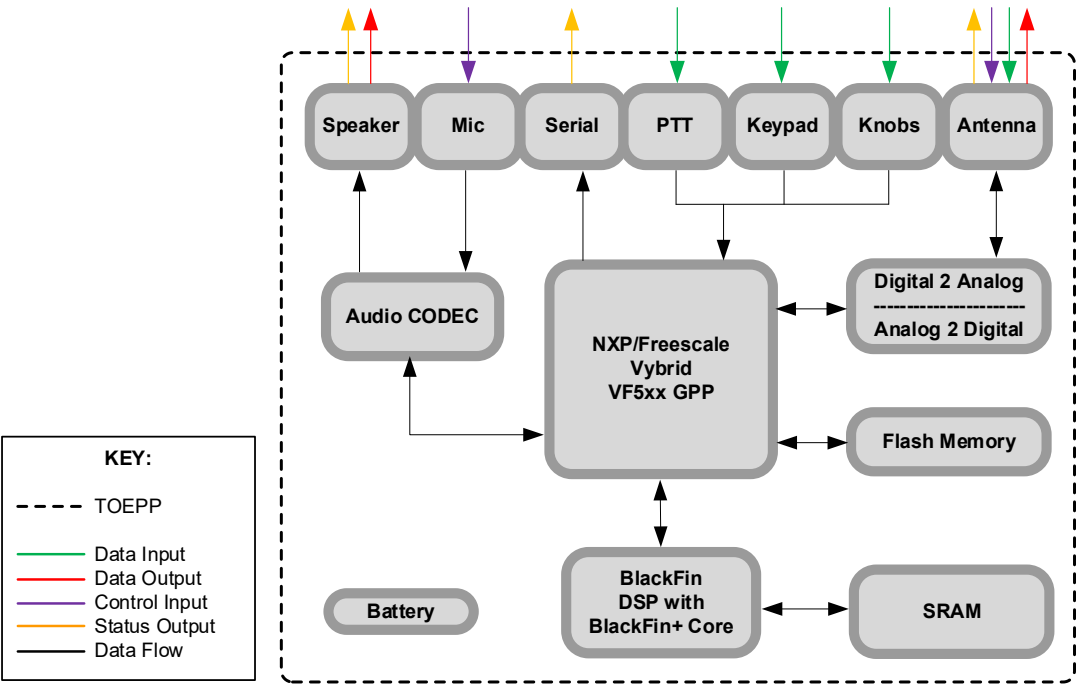


Figure 2. Radio Terminal Block Diagram

## 2.2 Tested and Vendor Affirmed Module Version and Identification

### 2.2.1 Tested Module Identification – Hardware

The module is not a hardware module.

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

The module has the following executable code sets shown in the table below.

| Package or File Name | Software/ Firmware Version | Features | Integrity Test                |
|----------------------|----------------------------|----------|-------------------------------|
| HALM.bin             | R07A01                     |          | 256-bit AES-CMAC verification |

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

### 2.2.3 Tested Module Identification – Hybrid Disjoint Hardware

The module is not a hybrid module.

## 2.2.4 Tested Operational Environments – Software, Firmware, Hybrid

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

| Operating System         | Hardware Platform | Processors         | PAA/PAI | Hypervisor or Host OS | Version(s) |
|--------------------------|-------------------|--------------------|---------|-----------------------|------------|
| L3Harris BIOS Kernel 1.0 | L3Harris XL-200P  | Blackfin BF707 DSP | No      | N/A                   | R07A01     |

**Table 3: Tested Operational Environments - Software, Firmware, Hybrid**

## 2.2.5 Vendor-Affirmed Operational Environments – Software, Firmware, Hybrid

Module operators may perform post-validation porting of the module and affirm the module's continued validation compliance. The table below specifies the vendor-affirmed operational environments claimed.

| Operating System         | Hardware Platform |
|--------------------------|-------------------|
| L3Harris BIOS Kernel 1.0 | XL-400P           |
| L3Harris BIOS Kernel 1.0 | XL-185P           |
| L3Harris BIOS Kernel 1.0 | XL-150P           |
| L3Harris BIOS Kernel 1.0 | XL-95P            |
| L3Harris BIOS Kernel 1.0 | XL-45P            |
| L3Harris BIOS Kernel 1.0 | XL-200M           |
| L3Harris BIOS Kernel 1.0 | XL-185M           |
| L3Harris BIOS Kernel 1.0 | XL-85M            |
| L3Harris BIOS Kernel 1.0 | XL-45M            |

**Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid**

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.

## 2.3 Excluded Components

The module does not exclude any components from the requirements.

## 2.4 Modes of Operation

### 2.4.1 Modes List and Description

The table below lists the modes of operation supported by the module.

| Mode Name | Description                                                     | Type     | Status Indicator                                                                                                                                                                                                                                                                                                              |
|-----------|-----------------------------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Approved  | When all pre-operational self-tests have successfully completed | Approved | Return from the HALM_SEND_STATUS message of any of the following states indicates that the module is operating in its Approved mode of operation: <ul style="list-style-type: none"> <li>• STATE_SELF_TEST</li> <li>• STATE_USER_MODE</li> <li>• STATE_KEY_ENTRY</li> <li>• STATE_CRYPT_OFR</li> <li>• STATE_ERROR</li> </ul> |

**Table 5: Modes List and Description**

## 2.5 Algorithms

### 2.5.1 Approved Algorithms

The module employs cryptographic algorithm implementations from the following sources:

- L3Harris AES Load Module version R07A01 (Cert. [A7422](#))

The module implements the Approved algorithms listed in the table below.

| Algorithm | CAVP Cert | Properties                                                                                                                                                                                      | Reference  |
|-----------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| AES-CMAC  | A7422     | Direction - Generation, Verification<br>Key Length - 256<br>MAC Length - MAC Length: 16-128 Increment 16, MAC Length: 8-128 Increment 8<br>Message Length - Message Length: 0-98296 Increment 8 | SP 800-38B |
| AES-ECB   | A7422     | Direction - Decrypt, Encrypt<br>Key Length - 128, 256                                                                                                                                           | SP 800-38A |
| AES-KW    | A7422     | Direction - Decrypt, Encrypt<br>Cipher - Cipher<br>Key Length - 256<br>Payload Length - Payload Length: 256                                                                                     | SP 800-38F |
| AES-OFB   | A7422     | Direction - Decrypt, Encrypt<br>Key Length - 256                                                                                                                                                | SP 800-38A |

**Table 6: Approved Algorithms**

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. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the table above are used by an Approved service of the module.

### 2.5.2 Vendor Affirmed Algorithms

The module does not implement any vendor-affirmed algorithms.

N/A for this module.

### 2.5.3 Non-Approved, Allowed Algorithms

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

N/A for this module.

### 2.5.4 Non-Approved, Allowed Algorithms with No Security Claimed

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

N/A for this module.

### 2.5.5 Non-Approved, Not Allowed Algorithms

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

N/A for this module.

## 2.6 Security Function Implementations

The table below lists the security function implementations for this module.

| Name                            | Type             | Description                                                          | Properties             | Algorithms                                                                       |
|---------------------------------|------------------|----------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------|
| AES for Symmetric Encryption    | BC-UnAuthEncrypt | AES (using ECB or OFB mode) for symmetric encryption of data         | Publication:SP 800-38A | AES-ECB: (A7422)<br>Direction: Encrypt<br>AES-OFB: (A7422)<br>Direction: Encrypt |
| AES for Symmetric Decryption    | BC-UnAuthDecrypt | AES (using ECB or OFB mode) for symmetric decryption of data         | Publication:SP 800-38A | AES-ECB: (A7422)<br>Direction: Decrypt<br>AES-OFB: (A7422)<br>Direction: Decrypt |
| AES-CMAC for Firmware Integrity | MAC              | AES (using CMAC mode) for the verification of the firmware image MAC | Publication:SP 800-38B | AES-CMAC: (A7422)<br>Direction: Verification                                     |
| AES-KW for Key Encryption       | BC-AuthEncrypt   | AES (using KW mode) for key encryption                               | Publication:SP 800-38F | AES-KW: (A7422)<br>Direction: Encrypt                                            |
| AES-KW for Key Decryption       | BC-AuthDecrypt   | AES (using KW mode) for key decryption                               | Publication:SP 800-38F | AES-KW: (A7422)<br>Direction: Decrypt                                            |

**Table 7: Security Function Implementations**

## 2.7 Algorithm Specific Information

### 2.7.1 KTS

The module does not establish SSPs using Approved key transport schemes (KTS). However, it does offer Approved algorithms that can be used by an external operator/application as part of an Approved KTS. Refer to section 2.10.2 below for additional details.

## 2.8 RBG and Entropy

The module does not have an internal DRBG or entropy source. All keys are generated externally and input into the module in encrypted form.

## 2.9 Key Generation

The module does not support key generation.

## 2.10 Key Establishment

The cryptographic module provides the cryptographic primitives necessary to support key agreement schemes and key transport methods for use by the calling application to establish keys.

### 2.10.1 Key Agreement Information

The module does not support key agreement methods.

### 2.10.2 Key Transport Information

The module offers as a service to a calling application CAVP-tested algorithms specified in the list below without establishing a key/SSP to be used by the module for cryptographic protection.

- AES-KW
- AES-CMAC

## 2.11 Industry Protocols

The module does not offer support for any industry protocols.

## 3. Cryptographic Module Interfaces

---

### 3.1 Ports and Interfaces

The module supports the following logical interfaces:

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

As a firmware library, the cryptographic module has no direct access to any of the host platform's physical ports; it communicates only to the DSP application, and communications from the DSP application are accomplished via the module's well-defined API. The table below contains a mapping of the logical interfaces supported by the module.

| Physical Port | Logical Interface(s) | Data That Passes                                                                                                                                                                              |
|---------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| N/A           | Data Input           | Includes content passed from the DSP application to the module to be encrypted/decrypted                                                                                                      |
| N/A           | Data Output          | Includes encrypted/decrypted content passed from the module to the DSP application                                                                                                            |
| N/A           | Control Input        | Includes a message ID (which gets translated by the module into the HALM API) and additional data to be used by the function (such as key size) passed from the DSP application to the module |
| N/A           | Status Output        | Includes status codes returned to the DSP application indicating the state of the module                                                                                                      |
| Power         | Power                | Power                                                                                                                                                                                         |

**Table 8: Ports and Interfaces**

The module does not output control information and has no specified control output interface.

## 4. Roles, Services, and Authentication

### 4.1 Authentication Methods

The module does not support authentication mechanisms.

### 4.2 Roles

The module supports the roles listed in the table below.

| Name           | Type | Operator Type  | Authentication Methods |
|----------------|------|----------------|------------------------|
| Crypto Officer | Role | Crypto Officer |                        |
| User           | Role | User           |                        |

**Table 9: Roles**

Roles are implicitly assumed by the calling application based on the service selected. The calling application is the module's only operator, and the module does not support multiple concurrent operators.

### 4.3 Approved Services

Descriptions of the available services are provided in the table below. Access rights for each SSP are specified 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.

| Name                            | Description                                                      | Indicator | Inputs   | Outputs                                                   | Security Functions | SSP Access     |
|---------------------------------|------------------------------------------------------------------|-----------|----------|-----------------------------------------------------------|--------------------|----------------|
| Show Status<br>HALM_SEND_STATUS | Returns the status of the last function called from the HALM_API | N/A       | API call | Status code of the last function called from the HALM_API |                    | Crypto Officer |

| Name                         | Description                                                                                                                                                                                                                                                | Indicator                                 | Inputs                                                                                   | Outputs                                                                                                                      | Security Functions              | SSP Access                          |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-------------------------------------|
| Self-Test<br>HALM_INITIALIZE | Allocates memory from the heap for use by the module, initiating the execution of the module's CASTs and integrity test<br><br>(The service is called by the DSP application's pre_main function when the application and module are loaded for execution) | N/A                                       | API call                                                                                 | Status code:<br>· 0x01 (HALM_INITIALIZE_OK)<br>· 0x81 (HALM_INIT_INTEGRITY_TEST_FAILED)<br>· 0x82 (HALM_INIT_AES_KAT_FAILED) | AES-CMAC for Firmware Integrity | Crypto Officer                      |
| HALM_LOAD_KEY                | Passes in the plaintext AES key to be used in subsequent block encryption processing                                                                                                                                                                       | Global status indicator + Status code     | API call; key length; key                                                                | Status code:<br>· 0x03 (HALM_LOAD_KEY_OK)                                                                                    |                                 | User - AES Key: W - AES-CMAC Key: W |
| HALM_GEN_PRIVATE_MI          | Generates a Message Indicator (MI) from the Initialization Vector (IV) value specified in the data input buffer                                                                                                                                            | Global status indicator + API status code | API call; input IV                                                                       | Status code:<br>· 0x04 (HALM_GEN_PRIVATE_MI_OK)                                                                              |                                 | User                                |
| HALM_GEN_KEYSTREAM           | Generates one or more blocks of Keystream data and adds the data to its internal Keystream buffer                                                                                                                                                          | Global status indicator + API status code | API call; iteration count                                                                | Status code:<br>· 0x05 (HALM_GEN_KEYSTREAM_OK)<br>· 0x85 (HALM_GEN_KEYSTREAM_OVERFLOW)                                       |                                 | User                                |
| HALM_P25_XOR                 | Performs logical Exclusive-Or (XOR) operation of P25 Voice or Data plaintext/ciphertext octets for AES-OFB encryptions/decryptions                                                                                                                         | Global status indicator + API status code | API call; number of octets; keystream table starting offset; plaintext/ciphertext octets | XOR'ed plaintext/ciphertext octets<br><br>Status code:<br>· 0x06 (HALM_P25_XOR_OK)<br>· 0x86 (HALM_P25_XOR_KEYSTREAM_ERROR)  |                                 | User                                |

| Name                        | Description                                                                                                                                                                             | Indicator                                    | Inputs                                                                            | Outputs                                                                                     | Security Functions                                           | SSP Access             |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------|------------------------|
| HALM_AES_OFB_PASS_THRU      | Encrypts one or more blocks of plaintext (or decrypts one or more blocks of ciphertext)                                                                                                 | Global status indicator or + API status code | API call; number of iterations; plaintext/ciphertext blocks to encrypt or decrypt | Encrypted/decrypted blocks<br><br>Status code:<br>· 0x07 (HALM_AES_OFB_PASSTHRU_OK)         | AES for Symmetric Encryption<br>AES for Symmetric Decryption | User - AES Key: E      |
| HALM_WRAP_KEY               | Wraps a temporary key generated by the DSP and passes it to the DSP in support of OTAR operations by the DSP                                                                            | Global status indicator or + API status code | API call; key encryption key; IV; key to be wrapped                               | Wrapped key<br><br>Status code:<br>· 0x0B (HALM_WRAP_KEY_OK)                                | AES-KW for Key Encryption                                    | User - AES-KW Key: W,E |
| HALM_UNWRAP_KEY             | Unwraps a wrapped key and passes it to the DSP in support of OTAR operations by the DSP                                                                                                 | Global status indicator or + API status code | API call; key encryption key; IV; wrapped key                                     | Unwrapped key<br><br>Status code:<br>· 0x0A (HALM_UNWRAP_OK)<br>· 0x8A (HALM_UNWRAP_FAILED) | AES-KW for Key Decryption                                    | User - AES-KW Key: W,E |
| HALM_LOAD_TEST_IV           | Loads an initialization vector (IV) into an internal HALM buffer for use                                                                                                                | Global status indicator or + API status code | API call; IV value                                                                | Status code:<br>· 0x0C (HALM_LOAD_TEST_IV_OK)                                               |                                                              | User                   |
| HALM_AES_ECB                | Encrypts each block of the plaintext directly and independently; used for P25 link layer authentication; used as part of a MAC generation algorithm implemented in the DSP for P25 OTAR | Global status indicator or + API status code | API call; key length; plaintext block                                             | ciphertext block<br><br>Status code:<br>· 0x0E (HALM_AES_ECB_OK)                            | AES for Symmetric Encryption                                 | User - AES Key: E      |
| Show versioning information | Returns the name and version of the module                                                                                                                                              | N/A                                          | API call                                                                          | Status output                                                                               |                                                              | Crypto Officer         |
| Cleanup                     | Deallocates memory                                                                                                                                                                      | N/A                                          | API call                                                                          | Status output                                                                               |                                                              | Crypto Officer         |

| Name    | Description                                        | Indicator | Inputs        | Outputs | Security Functions | SSP Access                                                                                                        |
|---------|----------------------------------------------------|-----------|---------------|---------|--------------------|-------------------------------------------------------------------------------------------------------------------|
| Zeroize | Zeroizes keys in volatile memory via manual reboot | N/A       | Manual reboot |         |                    | Crypto Officer - AES Key: Z - AES-CMAC Key: Z - AES-KW Key: Z User - AES Key: Z - AES-CMAC Key: Z - AES-KW Key: Z |

Table 10: Approved Services

The module does not support a non-Approved mode of operation. Thus, as allowed by *FIPS 140-3 IG 2.4.C*, the module provides indicators for the use of Approved security services through a combination of an explicit indication (via a global Approved mode indicator) and an implicit indication (via the successful completion of the service).

## 4.4 Non-Approved Services

The module does not offer any non-Approved services.

N/A for this module.

## 4.5 External Software/Firmware Loaded

The module does not support the loading of software/firmware from external sources.

## 4.6 Additional Information

The processing of specific HALM\_API commands may be prohibited while the module is in a particular state. Should the module receive a command that is prohibited in its current state, it will return the status code 0xFE (HALM\_REQUEST\_DENIED). If the module is in the STATE\_LOCKED state, the module responds with the current Status Word (state and status code) for any valid message it receives. In addition, if the module receives a HALM\_API command that it does not recognize, it will return the status code 0xFF (HALM\_INVALID\_MSG).

## 5. Software/Firmware Security

---

### 5.1 Integrity Techniques

All firmware components within the cryptographic boundary are verified using Approved integrity techniques implemented within the cryptographic module itself. The module compares a 256-bit AES-CMAC digest calculated at run-time with the 256-bit AES-CMAC digest computed at build-time. If the integrity tests fail, the module enters 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 key and the MAC digest are embedded in the module binary.

### 5.2 Initiate on Demand

The operator can initiate the pre-operational integrity test and conditional CASTs on demand by restarting the host radio terminal (thereby restarting the module).

## 6. Operational Environment

---

### 6.1 Operational Environment Type and Requirements

The Harris AES Load Module is a firmware cryptographic library that executes in a Non-Modifiable operational environment.

The cryptographic module has control over its own SSPs. The process and memory management functionality of the host terminal'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, thereby 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 environments. Processes that are spawned by the module are owned by the module and are not owned by external processes/operators.

## 7. Physical Security

---

The module executes on L3Harris radio terminal hardware meeting level 1 physical security requirements. All components of the hardware are made of production-grade materials, and all integrated circuits are coated with commercial standard passivation.

## 8. Non-Invasive Security

---

This section is not applicable. There are currently no approved non-invasive mitigation techniques references in Annex F of *ISO/IEC 19790:2012*.

## 9. Sensitive Security Parameters Management

### 9.1 Storage Areas

There is no mechanism within the module's cryptographic boundary for the persistent storage of SSPs. SSPs are held in volatile RAM during module operation.

The table below lists sensitive security parameters (SSPs) storage areas for this module. Section 9.4 selects from the storage areas listed and specifies the appropriate storage area in the "Storage" column if applicable to a given SSP.

| Storage Area Name | Description            | Persistence Type |
|-------------------|------------------------|------------------|
| RAM               | SSPs stored in the RAM | Dynamic          |

**Table 11: Storage Areas**

### 9.2 SSP Input-Output Methods

The table below lists SSP input and output methods for this module. Section 9.4 selects from the input and output methods listed and specifies the appropriate method in the "Inputs/Outputs" column if applicable to a given SSP.

| Name                         | From     | To  | Format Type | Distribution Type | Entry Type | SFI or Algorithm |
|------------------------------|----------|-----|-------------|-------------------|------------|------------------|
| External to RAM in plaintext | External | RAM | Plaintext   | Automated         | Electronic |                  |

**Table 12: SSP Input-Output Methods**

### 9.3 SSP Zeroization Methods

The table below lists SSP zeroization methods for this module. Section 9.4 selects from the zeroization methods listed and specifies the appropriate method in the "Zeroization" column if applicable to a given SSP.

| Zeroization Method | Description                                                   | Rationale                                                                                  | Operator Initiation              |
|--------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------|----------------------------------|
| Power cycle        | Power cycling the module's host radio terminal zeroizes SSPs. | Power cycling the module's host radio terminal zeroizes SSPs, yielding them irretrievable. | Operator power cycles the module |

**Table 13: SSP Zeroization Methods**

### 9.4 SSPs

The module supports the keys and other SSPs listed in the tables below.

| Name         | Description                                                                   | Size - Strength                                                                                                          | Type - Category     | Generated By | Established By | Used By                                                      |
|--------------|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|---------------------|--------------|----------------|--------------------------------------------------------------|
| AES Key      | Symmetric key used for ECB and OFB encryption and decryption operations       | [AES-ECB]<br>128 or 256 bits<br><br>[AES-OFB]<br>256 bits -<br>[AES-ECB]<br>128 or 256 bits<br><br>[AES-OFB]<br>256 bits | Symmetric Key - CSP |              |                | AES for Symmetric Encryption<br>AES for Symmetric Decryption |
| AES-CMAC Key | Symmetric key used as input for CMAC generate/verify operations               | 256 bits - 256 bits                                                                                                      | MAC - CSP           |              |                | AES-CMAC for Firmware Integrity                              |
| AES-KW Key   | Symmetric key used as input into the key encryption and decryption operations | 256 bits - 256 bits                                                                                                      | Symmetric Key - CSP |              |                | AES-KW for Key Encryption<br>AES-KW for Key Decryption       |

**Table 14: SSP Table 1**

| Name         | Input - Output               | Storage       | Storage Duration                 | Zeroization | Related SSPs |
|--------------|------------------------------|---------------|----------------------------------|-------------|--------------|
| AES Key      | External to RAM in plaintext | RAM:Plaintext | Until the module is power cycled | Power cycle |              |
| AES-CMAC Key | External to RAM in plaintext | RAM:Plaintext | Until the module is power cycled | Power cycle |              |
| AES-KW Key   | External to RAM in plaintext | RAM:Plaintext | Until the module is power cycled | Power cycle |              |

**Table 15: SSP Table 2**

## 9.5 Additional Information

Maintenance, including protection and zeroization of any keys and CSPs that exist outside the module's cryptographic boundary, are the responsibility of the end-user.

# 10. Self-Tests

The module performs pre-operational self-tests and conditional self-tests. Pre-operational tests are performed when the cryptographic module is instantiated, before the module transitions to the operational state. Conditional self-tests are performed by the module during module operation when certain conditions exist.

The pre-operational integrity tests and cryptographic algorithm self-tests are performed automatically via a default entry point (DEP) 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.

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 pre-operational self-tests listed in the table below.

| Algorithm or Test       | Test Properties | Test Method        | Test Type       | Indicator                                                               | Details                                                                   |
|-------------------------|-----------------|--------------------|-----------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Firmware Integrity Test | 256-bit         | Firmware integrity | SW/FW Integrity | 0x01 (HALM_INITIALIZE_OK)<br><br>0x81 (HALM_INIT_INTEGRITY_TEST_FAILED) | Firmware integrity test of the binary verifying a 256-bit AES-CMAC digest |

**Table 16: Pre-Operational Self-Tests**

## 10.2 Conditional Self-Tests

The module performs the conditional self-tests listed in the table below.

| Algorithm or Test                        | Test Properties | Test Method | Test Type | Indicator                                                        | Details  | Conditions                                      |
|------------------------------------------|-----------------|-------------|-----------|------------------------------------------------------------------|----------|-------------------------------------------------|
| AES-ECB Encrypt KAT (A7422) - Encrypt    | 128-bit         | KAT         | CAST      | 0x01 (HALM_INITIALIZE_OK)<br><br>0x82 (HALM_INIT_AES_KAT_FAILED) | Encrypt  | At module start-up, prior to the integrity test |
| AES-ECB Decrypt KAT (A7422) - Decrypt    | 128-bit         | KAT         | CAST      | 0x01 (HALM_INITIALIZE_OK)<br><br>0x82 (HALM_INIT_AES_KAT_FAILED) | Decrypt  | At module start-up, prior to the integrity test |
| AES-CMAC Generate KAT (A7422) - Generate | 256-bit         | KAT         | CAST      | 0x01 (HALM_INITIALIZE_OK)<br><br>0x82 (HALM_INIT_AES_KAT_FAILED) | Generate | At module start-up, prior to the integrity test |
| AES-CMAC Verify KAT (A7422) - Verify     | 256-bit         | KAT         | CAST      | 0x01 (HALM_INITIALIZE_OK)<br><br>0x82 (HALM_INIT_AES_KAT_FAILED) | Verify   | At module start-up, prior to the integrity test |

**Table 17: Conditional Self-Tests**

## 10.3 Periodic Self-Test Information

Module operators can initiate self-tests on demand for periodic testing of the module. The tables below specify the self-tests that are performed when periodic self-tests are initiated by the module operator.

| Algorithm or Test       | Test Method        | Test Type       | Period    | Periodic Method |
|-------------------------|--------------------|-----------------|-----------|-----------------|
| Firmware Integrity Test | Firmware integrity | SW/FW Integrity | On Demand | Manually        |

**Table 18: Pre-Operational Periodic Information**

| Algorithm or Test                        | Test Method | Test Type | Period    | Periodic Method |
|------------------------------------------|-------------|-----------|-----------|-----------------|
| AES-ECB Encrypt KAT (A7422) - Encrypt    | KAT         | CAST      | On Demand | Manually        |
| AES-ECB Decrypt KAT (A7422) - Decrypt    | KAT         | CAST      | On Demand | Manually        |
| AES-CMAC Generate KAT (A7422) - Generate | KAT         | CAST      | On Demand | Manually        |
| AES-CMAC Verify KAT (A7422) - Verify     | KAT         | CAST      | On Demand | Manually        |

**Table 19: Conditional Periodic Information**

## 10.4 Error States

The table below describes the module's error states/conditions and error state recovery methods.

| Name           | Description                                                                                                                                                                                                   | Conditions                 | Recovery Method                                                        | Indicator    |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|------------------------------------------------------------------------|--------------|
| Critical Error | The module denies requests for service, returning the status code 0xFE (HALM_REQUEST_DENIED) to the DSP application while the error state persists. All data output via the data output interface is blocked. | Upon failing any self-test | The operator must restart the host radio terminal by power cycling it. | STATE_LOCKED |

**Table 20: Error States**

While in the critical error state, all data output via any data output interface is blocked. The only message that can be used while in this state is the `HALM_SEND_STATUS()` message, which will return the status code indicating the module's current state (`STATE_LOCKED`). If restarting the terminal does not clear the error state, the operator should contact L3Harris Technologies for support.

## 10.5 Operator Initiation of Self-Tests

The operator can initiate the module's start-up self-tests by restarting the host terminal, thereby executing the `HALM_INITIALIZE()` message. The `HALM_INITIALIZE()` message can also be used during module operation to initiate the module's start-up self-tests on demand.

# 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 Installation, Initialization, and Startup Procedures

The module is initialized by the DSP when the host radio terminal is powered on. When the DSP loads into memory for execution, its `pre_main()` function automatically executes the HALM's `HALM_INITIALIZE` routine to run the module's conditional CASTs, followed by the pre-operational integrity self-test. The module is not considered to be in an operational state until the integrity test has successfully completed. The module's services are only made available after the module performs and successfully passes its integrity test.

## 11.2 Administrator Guidance

The CO should monitor the module's status regularly. If any irregular activity is noticed, or if the module is consistently reporting errors, the CO should contact L3Harris customer support for assistance.

The following list provides additional guidance for the CO:

- The `HALM_SEND_STATUS` message can be used to determine the module's current mode of operation. Return of any of the following states indicates that the module is operating in its Approved mode of operation:
  - `STATE_SELF_TEST`
  - `STATE_USER_MODE`
  - `STATE_KEY_ENTRY`
  - `STATE_CRYPTO_OFR`
  - `STATE_ERROR`
- The `requestStatus()` function can be used to obtain the name and versioning information of the module. When used with the parameter "type = 0", the function will return a pointer to the version string. The version string will include the module name and version, which can be correlated with the module's validation record.

## 11.3 Non-Administrator Guidance

Users can only access the module's cryptographic functionalities that are available to them. The User should report to the CO if any irregular activity is noticed.

## 11.4 Design and Rules

By design, the module follows or enforces the following rules of operation:

- Operators do not have access to any cryptographic services prior to assuming an authorized role.
- The module performs all self-tests without any operator action required.
- The module inhibits data output during self-tests, zeroization, and error states.
- Status information output by the module does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
- There are no restrictions on which keys or SSPs are zeroized by the zeroization service.
- The module does not support a maintenance interface or role.
- The module does not support manual SSP establishment method.
- The module does not have any proprietary external input/output devices used for entry/output of data.
- The module does not store any plaintext CSPs
- The module does not output intermediate key values.
- The module does not provide bypass services or ports/interfaces.

## 11.5 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of performing the zeroization methods described in section 9.3 above. This will ensure that any SSPs in volatile memory are zeroized.

## 12. Mitigation of Other Attacks

---

The module does not claim to mitigate any attacks beyond the FIPS 140-3 Level 1 requirements for this validation. Therefore, per *ISO/IEC 19790:2021* section 7.12, requirements for this section are not applicable.

# Appendix A. Acronyms and Abbreviations

Table 21 provides definitions for the acronyms and abbreviations used in this document.

**Table 21. Acronyms and Abbreviations**

| Acronym | Definition                                                                                 |
|---------|--------------------------------------------------------------------------------------------|
| AES     | Advanced Encryption Standard                                                               |
| API     | Application Programming Interface                                                          |
| BIOS    | Basic Input/Output System                                                                  |
| CAST    | Cryptographic Algorithm Self-Test                                                          |
| CAVP    | Cryptographic Algorithm Validation Program                                                 |
| CBC     | Cipher Block Chaining                                                                      |
| CCCS    | Canadian Centre for Cyber Security                                                         |
| CMAC    | Cipher-Based Message Authentication Code                                                   |
| CMVP    | Cryptographic Module Validation Program                                                    |
| CO      | Cryptographic Officer                                                                      |
| codec   | Coder/Decoder                                                                              |
| CPU     | Central Processing Unit                                                                    |
| CSP     | Critical Security Parameter                                                                |
| CTR     | Counter                                                                                    |
| DEP     | Default Entry Point                                                                        |
| DRBG    | Deterministic Random Bit Generator                                                         |
| DSP     | Digital Signal Processor                                                                   |
| ECB     | Electronic Code Book                                                                       |
| FIPS    | Federal Information Processing Standard                                                    |
| GPP     | General Purpose Processor                                                                  |
| IG      | Implementation Guidance                                                                    |
| ISO/IEC | International Organization for Standardization / International Electrotechnical Commission |
| KAT     | Known Answer Test                                                                          |
| KTS     | Key Transport Scheme                                                                       |
| KW      | Key Wrap                                                                                   |
| MAC     | Message Authentication Code                                                                |
| MSI     | Module Software Interface                                                                  |
| NIST    | National Institute of Standards and Technology                                             |
| OFB     | Output Feedback                                                                            |
| OS      | Operating System                                                                           |

| Acronym | Definition                                          |
|---------|-----------------------------------------------------|
| PTT     | Push-To-Talk                                        |
| RAM     | Random Access Memory                                |
| RNG     | Random Number Generator                             |
| SP      | Special Publication                                 |
| SRAM    | Static Random Access Memory                         |
| SSP     | Sensitive Security Parameter                        |
| TOEPP   | Tested Operational Environment's Physical Perimeter |
| XOR     | Exclusive-Or                                        |

---

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](mailto:info@corsec.com)

Web: [www.corsec.com](http://www.corsec.com)

---