

MolecuLight Inc.

MolecuLightDX Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Version 1.0

Table of Contents

1 General	5
1.1 Overview	5
1.2 Security Levels	5
2 Cryptographic Module Specification	5
2.1 Description	5
2.2 Tested and Vendor Affirmed Module Version and Identification	6
2.3 Excluded Components	7
2.4 Modes of Operation	7
2.5 Algorithms	8
2.6 Security Function Implementations	9
2.7 Algorithm Specific Information	9
2.8 RBG and Entropy	10
2.9 Key Generation	10
2.10 Key Establishment	10
2.11 Industry Protocols	10
3 Cryptographic Module Interfaces	10
3.1 Ports and Interfaces	10
4 Roles, Services, and Authentication	11
4.1 Authentication Methods	11
4.2 Roles	11
4.3 Approved Services	11
4.4 Non-Approved Services	12
4.5 External Software/Firmware Loaded	12
5 Software/Firmware Security	13
5.1 Integrity Techniques	13
5.2 Initiate on Demand	13
6 Operational Environment	13
6.1 Operational Environment Type and Requirements	13
6.2 Configuration Settings and Restrictions	13
7 Physical Security	14
8 Non-Invasive Security	14
9 Sensitive Security Parameters Management	14
9.1 Storage Areas	14
9.2 SSP Input-Output Methods	14
9.3 SSP Zeroization Methods	14

9.4 SSPs	15
10 Self-Tests.....	15
10.1 Pre-Operational Self-Tests	15
10.2 Conditional Self-Tests.....	16
10.3 Periodic Self-Test Information.....	16
10.4 Error States	16
10.5 Operator Initiation of Self-Tests	17
11 Life-Cycle Assurance	17
11.1 Installation, Initialization, and Startup Procedures.....	17
11.2 Administrator Guidance	17
11.3 Non-Administrator Guidance.....	17
11.4 Design and Rules	17
11.6 End of Life	17
12 Mitigation of Other Attacks	18

List of Tables

Table 1: Security Levels	5
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)....	7
Table 3: Tested Operational Environments - Software, Firmware, Hybrid	7
Table 4: Modes List and Description	8
Table 5: Approved Algorithms	8
Table 6: Security Function Implementations.....	9
Table 7: Ports and Interfaces	10
Table 8: Roles.....	11
Table 9: Approved Services	12
Table 10: Storage Areas	14
Table 11: SSP Input-Output Methods.....	14
Table 12: SSP Zeroization Methods.....	15
Table 13: SSP Table 1	15
Table 14: SSP Table 2.....	15
Table 15: Pre-Operational Self-Tests	16
Table 16: Conditional Self-Tests	16
Table 17: Pre-Operational Periodic Information.....	16
Table 18: Conditional Periodic Information.....	16
Table 19: Error States	16

List of Figures

Figure 1: Block Diagram.....	6
------------------------------	---

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the MolecuLight Inc. MolecuLightDX Cryptographic Module, hereafter “the module”. It contains a specification of the security rules under which the module must operate and describes how the module meets the requirements as specified in FIPS 140-3 for a Security Level 1 Software cryptographic module.

1.2 Security Levels

Table 1 describes the security level of each section in this document. The module meets the overall requirements of FIPS 140-3 Security Level 1.

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The MolecuLightDX Cryptographic Module is a software module running on a non-modifiable operating environment (OE). The physical embodiment is provided by the multiple-chip standalone computing platform on which it is installed. The module provides Java-based cryptographic services to the dedicated MolecuLightDX applications on the host device via an application programming interface (API).

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Module Characteristics: N/A

Cryptographic Boundary:

The module's cryptographic boundary is delimited by the set of executable files that constitute the module. The executable files constituting the module are listed in Table 2.

Tested Operational Environment's Physical Perimeter (TOEPP):

The Tested Operating Environment (TOEPP) is delimited by the physical perimeter of the host device on which the module runs. The block diagram in Figure 1 depicts the module's cryptographic boundary (denoted via the dashed green line) and TOEPP.

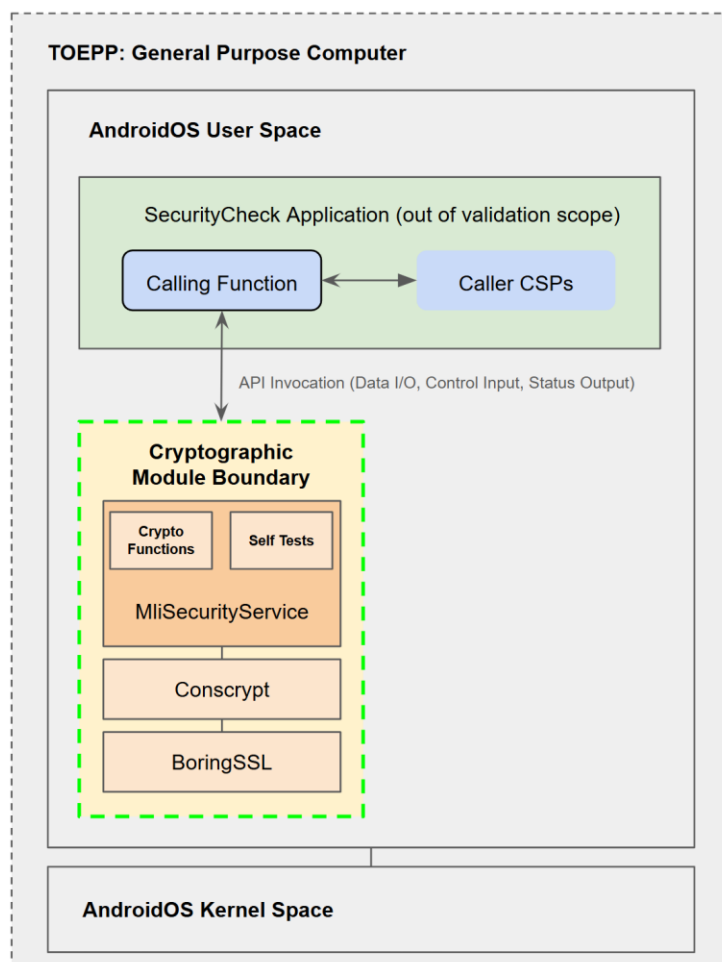


Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software / Firmware Version	Features	Integrity Test
/system/lib64/libcrypto.so, /system/lib/libcrypto.so, /system/lib64/libcrypto_utils.so, /system/lib/libcrypto_utils.so, /system/apex/com.android.vndk.current/lib/libcrypto.so, /system/apex/com.android.vndk.current/lib64/libcrypto.so, /system/apex/com.android.vndk.current/lib64/libcrypto_utils.so, /system/apex/com.android.vndk.current/lib/libcrypto_utils.so, /system/apex/com.android.conscrypt/lib64/libjavacrypto.so, /system/apex/com.android.conscrypt/lib/libjavacrypto.so, /system/apex/com.android.conscrypt/javalib/conscrypt.jar, /system/apex/com.android.art.release/javalib/core-oj.jar, /system/priv-app/MliSecurityServiceApp/MliSecurityServiceApp.apk, /system/framework/com.android.mlisecurityservice.jar, apkChecksumFile.dat	1.0.854f51	N/A	SHA2-256

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

The identifier “854f51” output by the Show Version Service corresponds to the module name MolecuLightDX Cryptographic Module.

Tested Operational Environments - Software, Firmware, Hybrid:

The module was tested on the following operating environments:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
MolecuLightDX-v11.1.25335	Qualcomm Open-Q™ 626 μSOM	Snapdragon 626 64-bit	No	N/A	1.0.854f51

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

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

There are no components within the cryptographic boundary excluded from the requirements of FIPS 140-3.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Types of Services	Status Indicator
Approved mode	Automatically entered whenever a service is requested	Approved	Equivalent to the indicator of the requested service as defined in section 4.3

Table 4: Modes List and Description

The module supports a single, approved mode of operation with only approved services. There are no non-approved modes, degraded modes, or non-approved services available to the module. The module will operate in the approved mode once it is powered on. No additional actions are required by the operator to invoke the approved mode.

2.5 Algorithms

Approved Algorithms:

The table below lists the approved algorithms supported by the module:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7783	Direction - Decrypt, Encrypt Key Length - 256	SP 800-38A
HMAC-SHA2-256	A7783	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
KDF SP800-108	A7783	KDF Mode - Counter MAC Mode - HMAC-SHA2-256 Supported Lengths - Supported Lengths: 256 Fixed Data Order - After Fixed Data Counter Length - 8 Supports Empty IV - No Requires Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
SHA2-256	A7783	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES Encryption	BC-UnAuthEncrypt	Encrypt a plaintext with AES	Standard:FIPS 197, NIST SP 800-38A	AES-CBC: (A7783) Key Length: 256
AES Decryption	BC-UnAuthDecrypt	Decrypt a ciphertext with AES	Standard:FIPS 197, NIST SP 800-38A	AES-CBC: (A7783) Key Length: 256
SHA	SHA	Compute a message digest	Standard:FIPS PUB 180-4	SHA2-256: (A7783) Message Length: 0-65536 Increment 8
Key Derivation	KBKDF	Key-Based Key Derivation Function	Standard:NIST SP 800-108r1	KDF SP800-108: (A7783) KDF Mode: Counter MAC Mode: HMAC-SHA2-256 Supported Lengths: 256 Fixed Data Order: After Fixed Data Counter Length: 8 Custom Key In Length: 0 HMAC-SHA2-256: (A7783) MAC: 256 Key Length: 256

Table 6: Security Function Implementations

2.7 Algorithm Specific Information

The module complies with IG C.L. No truncation of SHA2-256 is performed within any of the module's security services.

The module complies with IG D.M. An AES key generated through KDF SP 800-108 (Cert. #A7783) is derived from an AES key that has been entered using an approved method, namely manual electronic entry per IG 9.5.A. The encoding of the data fields (i.e., Label, Context, and Length L) is complaint to Section 6.4 of NIST SP 800-133r2. No asymmetric keys are generated through KBKDF. HMAC-SHA2-256 is used as the PRF with MAC output length of 256 bits and

key length of 256 bits, thus providing the strength of 256 per Section 6.1 of NIST SP 800-108r1-upd1.

The module does not truncate HMAC outputs.

2.8 RBG and Entropy

The module does not include an entropy source or DRBG.

2.9 Key Generation

The module does not support any key generation methods.

2.10 Key Establishment

The module offers key derivation using KBKDF per NIST SP 800-108r1-upd1. The module implements KBKDF in counter mode with HMAC-SHA2-256. The key derivation key is input to the module via the module's API. The SSPs derived by the module are exported to the calling application and are not stored persistently in the module.

2.11 Industry Protocols

The module does not support any industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

As a software only module, the module does not have physical ports. The module has no access to or from the external ports of the host device on which it runs. All input and output is via the API of the module to the calling application, which is entirely within the boundary of the operating environment. The module's logical interfaces are defined by the module's API. The following table describes the module's interfaces:

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API data input parameters
N/A	Data Output	API output parameters
N/A	Control Input	API function calls, API control input parameters
N/A	Status Output	API return values

Table 7: Ports and Interfaces

The module's interfaces are logically separated by the module's API. All data output via the data output interface is inhibited when the module is performing pre-operational self-tests, zeroization, or when the module is in an error state.

The module does not support physical status indicators, does not support any manual controls used to physically enter control signals, and does not contain support for physical or electrical characteristics of intervening systems. The module does not support a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module is a FIPS 140-3 Security Level 1 software module that does not support authentication mechanisms.

4.2 Roles

The module supports the Crypto Officer role only. The Crypto Officer role is assumed implicitly upon module startup.

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

Table 8: Roles

The module does not support multiple concurrent operators.

4.3 Approved Services

The module supports the following approved services:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric Encryption	Encrypt sensitive data	Service completion	AES key, message	Encrypted message or null	AES Encryption	Crypto Officer - Derived AES Key: W,E
Symmetric Decryption	Decrypt sensitive data	Service completion	AES key, encrypted message	Decrypted message or null	AES Decryption	Crypto Officer - Derived AES Key: W,E
Hashing	Create message digest	Service completion	Message	Message digest or null	SHA	Crypto Officer
Key Derivation	Perform key-based key derivation	Service completion	Key	Key or null	Key Derivation	Crypto Officer - Reference Key: W,E - Derived AES Key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Self-Test	Run self-tests on demand	Service completion	None	True if all tests pass (Normal status). False (error status), otherwise	None	Crypto Officer
Show Status	Get the module status - one of: UNKNOWN, STARTUP, NORMAL, ERROR	Service completion	None	Module status - one of: UNKNOWN, STARTUP, NORMAL, ERROR	None	Crypto Officer
Show Version	Output the module identifier and version	Service completion	None	Module identifier/version or null	None	Crypto Officer
Power off / Zeroize	Power off the module and zeroize temporary SSPs stored in RAM	Service completion	None	None	None	Crypto Officer - Derived AES Key: Z - Reference Key: Z

Table 9: Approved Services

The module supports a Global approved service indicator for modules having approved services only per FIPS 140-3 IG 2.4.C example scenario 2. A non-null return value from the requested service indicates the successful completion of the service, while a null return value indicates an error. The successful completion of a service is an implicit indicator for the use of an approved service.

4.4 Non-Approved Services

The module does not provide any non-approved services.

4.5 External Software/Firmware Loaded

The module does not have the capability of loading software or firmware from an external source.

5 Software/Firmware Security

The module's software components are listed in Table 2. The module's software is in executable form.

5.1 Integrity Techniques

The Software Integrity Test is performed as part of the Pre-Operational Self-Tests. The module uses SHA2-256 as the integrity technique for all of the module's executable code. The module verifies the software integrity by computing the SHA2-256 digest of all software components comprising the module and comparing the computed digests with known values of the digests that were computed during module development. The known value of the SHA2-256 digest of the 'MliSecurityServiceApp.apk' file is stored in a separate file named 'apkChecksumFile.dat'. The known values of the SHA2-256 digests of all other files comprising the module are stored in the file 'MliSecurityServiceApp.apk'. If any of the computed SHA2-256 digests do not match the known values, the Software Integrity Test fails, and the module enters the error state.

The module performs a self-test on the SHA2-256 algorithm to ensure that it is running correctly before performing the Software Integrity Test. If the SHA2-256 self-test fails, the module enters the error state.

5.2 Initiate on Demand

The consuming application operator can initiate all the module's self-tests, including the Software Integrity Test, at any time using the "Self-Test" service.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The module executes in a modifiable operating environment. The single operational environment that the module supports is listed in Section 2.2 – Tested and Vendor Affirmed Module Version and Identification.

Each process running in the operational environment is logically separated from all other processes by the operating system and hardware. The Module functions entirely within a single process. The operational environment enforces process separation using virtual memory. The virtual memory system uses the memory management unit (MMU) hardware to prevent processes from accessing the memory of another process.

The module does not spawn any processes.

6.2 Configuration Settings and Restrictions

There are no security rules, settings, or restrictions on the configuration of the operational environment imposed by the module.

7 Physical Security

N/A for this module. The module is implemented purely in software.

8 Non-Invasive Security

N/A for this module. The module does not claim any non-invasive security measures.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The module does not persistently store any SSPs. SSPs used by the module are stored temporarily in RAM within the OE.

Storage Area Name	Description	Persistence Type
RAM	Temporary storage used by the module as part of service execution. This storage area only stores data while the operating environment is powered on.	Dynamic

Table 10: Storage Areas

9.2 SSP Input-Output Methods

SSPs are input to and output from the module via the module's API.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Plaintext Input	Outside the module	RAM	Plaintext	Manual	Electronic	
Plaintext Output	RAM	Outside the module	Plaintext	Manual	Electronic	

Table 11: SSP Input-Output Methods

Input and output of SSPs occur between the module and an application outside the module's boundary but within the TOEPP. This type of SSP input/output is classified as Manual Distribution/Electronic Entry per FIPS 140-3 IG 9.5.A according to the "CM Software to/from App via TOEPP Path" scenario in the CSP Establishment Table.

9.3 SSP Zeroization Methods

All SSPs are zeroized implicitly when the module is powered off.

Zeroization Method	Description	Rationale	Operator Initiation
Power-cycle	Zeroizes the volatile memory	All SSPs stored in volatile memory are overwritten when power is removed. Successfully power-cycling the module implicitly indicates that the zeroization is complete.	Power-cycle

Table 12: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Reference Key	AES key used for key derivation	256 bits - 256 bits	Symmetric Key - CSP	Externally		Key Derivation
Derived AES Key	AES key derived by KBKDF	256 bits - 256 bits	Symmetric Key - CSP		Key Derivation	AES Encryption AES Decryption

Table 13: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Reference Key	Plaintext Input Plaintext Output	RAM:Plaintext	Until power-cycling	Power-cycle	
Derived AES Key	Plaintext Input Plaintext Output	RAM:Plaintext	Until power-cycling	Power-cycle	

Table 14: SSP Table 2

10 Self-Tests

Cryptographic self-tests must be executed by the application upon startup and may be executed on demand. All self-tests must pass before the module can transition to an operational state where cryptographic operations are available. If any of the tests fail, the module will enter the error state and will no longer operate until all the tests are re-run successfully and pass.

Once the self-tests are performed, all data output via the data output interface is inhibited until the self-tests are completed.

10.1 Pre-Operational Self-Tests

The module performs one pre-operational self-test: the software integrity test, which uses SHA2-256. The details of the software integrity test are provided in Section 5.1.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Software Integrity Test	N/A	SHA2-256 hash	SW/FW Integrity	Return value	SHA2-256 hash

Table 15: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The module performs conditional cryptographic algorithm self-tests (CASTs) on all supported approved algorithms. All CASTs performed are Known Answer Tests (KATs). If any of the self-tests fail, the module will output an error indicator and enter the error state.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC Encrypt	Key length: 256 bits	KAT	CAST	Return value	Encrypt KAT	On startup and on-demand
AES-CBC Decrypt	Key length: 256 bits	KAT	CAST	Return value	Decrypt KAT	On startup and on-demand
SHA2-256	N/A	KAT	CAST	Return value	SHA2-256 KAT	On startup and on-demand
KBKDF	KDF Mode: Counter	KAT	CAST	Return value	KBKDF KAT	On startup and on-demand

Table 16: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Software Integrity Test	SHA2-256 hash	SW/FW Integrity	On startup and on-demand	Invocation of Self-Test service

Table 17: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Encrypt	KAT	CAST	On-demand	Invocation of Self-Test service
AES-CBC Decrypt	KAT	CAST	On-demand	Invocation of Self-Test service
SHA2-256	KAT	CAST	On-demand	Invocation of Self-Test service
KBKDF	KAT	CAST	On-demand	Invocation of Self-Test service

Table 18: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	The module enters the error state if one or more of the self-tests fail	Self-test failure	Restart the application or re-attempt the self-tests on demand	Show Status service indicates ERROR state

Table 19: Error States

10.5 Operator Initiation of Self-Tests

The module provides a “Self-Test” service to perform the module’s pre-operational and conditional self-tests on demand. During the execution of the on-demand self-tests, services are unavailable and data output is inhibited.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

No additional steps are required for the installation of the module as it is integrated with the AOS Operating Environment during the build process and manufacture of the device. Applications are configured to run the module’s self-tests before the module can transition to the operational state where security module functions can be used.

11.2 Administrator Guidance

All the approved services and ports and interfaces described in this document are available to the administrator (Crypto Officer) role.

11.3 Non-Administrator Guidance

The module supports the Crypto Officer role only, so non-administrator guidance is not applicable for this module.

11.4 Design and Rules

The module enforces the following security rules:

- The cryptographic module does not support a non-approved mode of operation and only operates in an approved mode of operation.
- The data output interface is inhibited during self-tests, zeroization, and when the module is in the error state.
- When the cryptographic module is in an error state, it ceases to provide cryptographic services, inhibits all data outputs, and provides status of the error.
- The cryptographic module does not support multiple concurrent operators.
- The cryptographic module protects CSPs from unauthorized disclosure, unauthorized modification, and unauthorized substitution.
- The cryptographic module does not support bypass capability and does not implement bypass tests.
- The cryptographic module does not support a maintenance role or maintenance access interface.
- The cryptographic module is not a radio and does not support any wireless interfaces or OTAR.

11.6 End of Life

No SSPs are maintained within the security module. When a host device running the module is decommissioned, the operator may remove and destroy the internal SDCARD and/or reformat the internal device storage.

12 Mitigation of Other Attacks

The module is not designed to mitigate other attacks beyond the scope of the FIPS 140-3 Security Level 1 requirements.