Cord3 Innovation Inc
Cord3 Cryptographic Module
v1.1

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.2.1
Date:4th February, 2019

Cord3 Innovation Inc
900 Morrison Drive, Suite 206
Ottawa, Ontario
Canada K2H 8K7
www.cord3inc.com

Table of Contents

List of Tables

List of Figures

Acknowledgements

The Cord3 Cryptographic Module is a rebranding of the OpenSSL FIPS Object Module targeted and tested for specific Cord3 Innovation platforms. This document was derived from the security policy associated with the OpenSSL FIPS 140-2 validation certificate number 2398.

References

Reference	Full Specification Name
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[FIPS STD]	FIPS PUB 140-2, Security Requirements for Cryptographic Modules, May 25, 2001, CHANGE NOTICES (12-03-2002) http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
[FIPS DTR]	Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules, January 4, 2011 Draft http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf
[FIPS IG]	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, April 25, 2014 http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

Vendor Name References

[SP] Cord3 Cryptographic Module Security Policy

1 Introduction

1.1 Purpose

This non-proprietary Security Policy for the Cord3 cryptographic module by Cord3 Innovation describes how the module meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode.

This document was prepared as part of the Level 1 FIPS 140-2 validation of the module. The following table lists the module's FIPS 140-2 security level for each section.

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

Table 1 - FIPS 140-2 Sections and Security Levels

The Module's software version for this validation is 1.1 rebranded and based on the OpenSSL FIPS Object SE Module (certificate 2398).

1.2 Background

Federal Information Processing Standards Publication (FIPS PUB) 140-2 – *Security Requirements for Cryptographic Modules* details the requirements for cryptographic modules. More information on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP), the FIPS 140-2 validation process, and a list of validated cryptographic modules can be found on the CMVP website:

http://csrc.nist.gov/groups/STM/cmvp/index.html

More information about Cord3 Innovation Unity product, can be found on the Cord3 website:

http://cord3inc.com/

1.3 Document Organization

This non-proprietary Security Policy is part of the Cord3 Cryptographic Module software FIPS 140-2 submission package. Other documentation in the submission package includes:

- Product documentation
- Vendor evidence documents
- Finite state model
- Additional supporting documents

The Cord3 Cryptographic Module is also referred to in this document as the cryptographic module, or the module.

2 Module Overview

The Cord3 Unity product provides Data Centric Security (DCS) - the DCS architecture defines the next generation of information security through its ability to address the following four basic challenges in information assurance.

1. The specification, application and enforcement of a unified and holistic security policy across all information assets;
2. The restriction of transactions against information assets to only those communities with the policy right to perform those actions;
3. The ability to provide assurance that information is only released to those users that have a policy right to access it; and
4. The use of a trusted audit facility that records the details related to the release of information in a tamper-resistant form.

The Cord3 Cryptographic Module makes use of the OpenSSL EVP_* API calls, namely:

EVP Cipher Routines

- EVP_CIPHER_CTX_init();
- EVP_CIPHER_CTX_key_length()
- EVP_CIPHER_CTX_iv_length()
- EVP_CIPHER_CTX_cleanup()
- EVP_CipherInit_ex()
- EVP_CipherUpdate()
- EVP_CipherFinal_ex()

Cipher Listings:

- EVP_aes_128_cbc()
- EVP_aes_192_cbc()
- EVP_aes_256_cbc()

These DCS capabilities are delivered as a security overlay to existing application, network and information architectures. An overview of the DCS cryptographic service is provided in Figure 1.

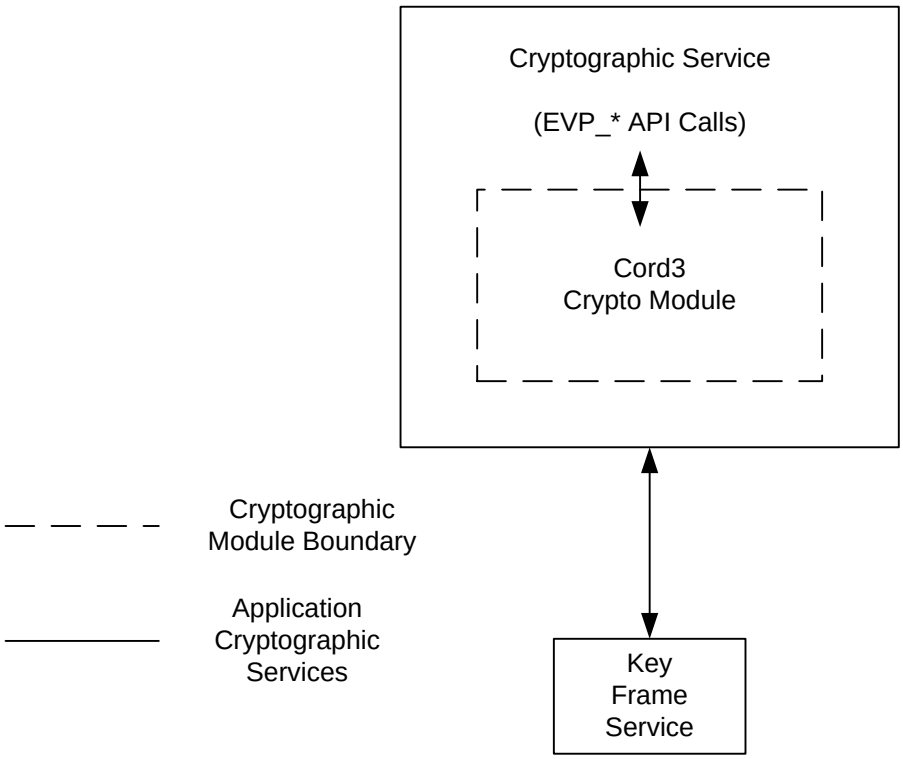

Figure 1: DCS Cryptographic Service Overview

2.1 Module Specification

The cryptographic module is contained within a general purpose computer and is a software module, multichip standalone module embodiment. The physical cryptographic boundary is the general purpose computer on which the module is installed. The logical cryptographic boundary of the Module is the fipscanister object module, a single object module file named fipscanister.o (Linux/Unix)

The product Cryptographic Services call a reduced set of cryptographic library functions in terms of cryptographic algorithms and key lengths, which are required by the product calling application to cryptographically protect data assets.

The cryptographic logical boundary is shown in Figure 2.

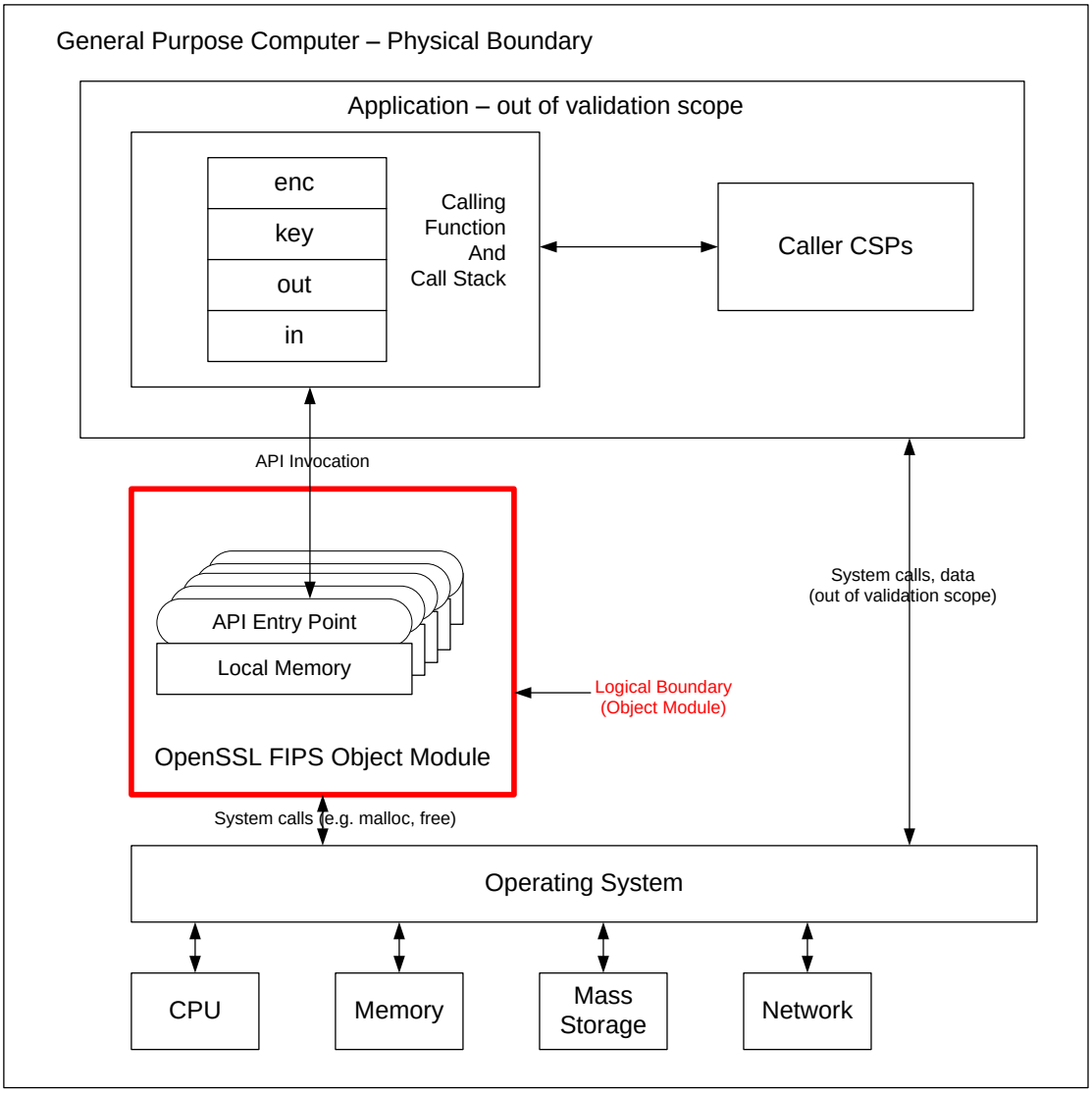

Figure 2: Logical Boundary

2.2 Ports and Interfaces

The physical ports of the Module are the same as the computer system on which it is executing. The logical interface is a C language application program interface (API).

As a software module, the module's manual controls, physical indicators, and physical and electrical characteristics are those of the host platform.Control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in error state, all output on the logical data output interface is inhibited. The module is singlethreaded and in error scenarios returns only an error value (no data output is returned). The module's logical interfaces are as follows:

Table 2 describes the module's logical interface mapping as defined in FIPS 140-2.

FIPS-140-2 Interface	Physical Interface	Logical Interface	Description
Data Input	Network port, Serial port, USB port, SCSI/SATA Controller	Input parameters to API calls	The API calls that accept input data for processing through their arguments.
Data Output	Network port, Serial port, USB port, SCSI/SATA Controller	Output parameters fromAPI calls	The API calls that return by means of their return codes or argument generated or processed data back to the caller.
Control Input	Network port, Serial port, USB port, Power button	API Function calls	The API calls that are used to initialize and control the operation of the module.
Status Output	Network port, Serial port, USB port, Graphics controller	Return values from API calls	Return values for API calls; Module generated error messages
Power Input	General purpose computer power	Not Applicable	Not Applicable

Table 2 – Logical Interface Mapping

2.3 Roles & Services

2.3.1 Roles

The Module implements the required User and Crypto Officer roles and does not require authentication for those roles. Only one role may be active at a time, the Module does not allow concurrent operators.

Both roles have access to all of the services provided by the Module.

- User Role (User): Loading the Module and calling any of the API functions.
- Crypto Officer Role (CO): Installation of the Module on the host computer system and calling any API functions.

Each role and their corresponding services are detailed in the sections below. Please note that the keys and Critical Security Parameters (CSPs) listed in Table 3 indicates the types of access required using the following notation:

- R – Read: The CSP is read.

- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an FIPS-Approved or Allowed security function or authentication mechanism.

2.3.2 Services

All services implemented by the Module are listed in Table 3, along with a description of service Critical Security Parameters (CSP) access.

Service	Operator	Description	Input	Output	CSP
Initialize	CO	Module initialization	Refer to Section 2.6.3	Refer to Section 2.6.4	Does not access CSPs.
Self-test	User, CO	Perform self tests (FIPS_selftest).	Refer to Section 2.6.3	Refer to Section 2.6.4	Does not access CSPs
Show status	User, CO	Show status of the module state	Refer to Section 2.6.3	Refer to Section 2.6.4	Does not access CSPs
Zeroize	User, CO	Zeroizes all CSPs	Refer to Section 2.6.3	Refer to Section 2.6.4	All CSPs – W
Symmetric encrypt/decrypt	User, CO	Encrypt or decrypt data using supplied key and algorithm specification (key passed in by the calling process)	Yes	No	AES EDK – RX
Message digest	User, CO	Used to generate a SHA1 or SHA2 message digest.	Refer to Section 2.6.3	Refer to Section 2.6.4	Does not access CSPs.
Keyed Hash	User, CO	Used to generate or verify data integrity with HMAC. Executes using HMAC Key (passed in by the calling process).	No	No	HMAC Key - RX

Table 3 –Services and CSP Access

2.4 Physical Security

The Cord3 Cryptographic Module is a software module, which FIPS defines as a multi-chip standalone cryptographic module. As such, it does not include physical security mechanisms. Thus, the FIPS 140-2 requirements for physical security are not applicable.

2.5 Tested Configurations

The module was tested and found to be compliant with FIPS 140-2 requirements on the following platforms:

- Eurocom D900F using Intel Xeon W3680 processor w/CentOS 7 64 bit
- Eurocom D900F using Intel Xeon W3680 processor w/CentOS 7 64 bit (without Processor Algorithm Accelerator -PAA)
- Dell PowerEdge R430 using Intel Xeon E5-2623v3 processor w/CentOS 7 64 bit on VMware ESXi 6.5

- Dell PowerEdge R430 using Intel Xeon E5-2623v3 processor w/CentOS 7 64 bit on VMware ESXi 6.5 (without PAA)

In addition to its full AES software implementations, the Cord3 Cryptographic Module is capable of leveraging the AES-NI instruction set of supported Intel and AMD processors in order to accelerate AES calculations.

All cryptographic keys and CSPs are under the control of the OS, which protects its CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to CSPs through its well-defined API.

The tested operating systems segregate user processes into separate process spaces. Each process space is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the calling application, and implicitly satisfies the FIPS 140-2 requirement for a single user mode of operation.

2.6 Modes of Operation and Cryptographic Functionality

2.6.1 Algorithm Implementations

The module supports the CSPs listed below in Table 4 and 5.

A list of FIPS-Approved algorithms implemented by the module can be found in Table 4.

Algorithm	Modes and Key Sizes	Cert Number	
		Non Virtual Machine	Virtual Machine
AES (PAA)	AES CBC and ECB (128/192/256) encrypt / decrypt key	5575	5517
AES (Non PAA)	AES CBC and ECB (128/192/256) encrypt / decrypt key	5686	5685
HMAC	HMAC-SHA-1, HMAC-SHA-256	3716	3673
SHA	SHA-1, SHA-256	4479	4428

Table 4 - FIPS-Approved Algorithm Implementations

The module does not output intermediate key generation values.

2.6.2 Critical Security Parameters

CSP N	Usage	Storage	Generation	Input	Output	Zeroization	Access
AES EDK	Encryption and decryption of data	RAM	N/A	Yes	No	Power off	CO, User
HMAC Integrity Key	Module integrity	RAM, Disk	N/A	No	No	No	CO

Table 5 - Cryptographic Keys, Key Components, and CSPs

2.6.3 Key Generation & Input

All CSPs enter the module's logical boundary in plaintext as API parameters, associated by memory location. However, none cross the physical boundary.

2.6.4 Key Output

The module does not output CSPs and none cross the physical boundary.

2.6.5 Storage

For all CSPs the RAM memory is associated by memory location. The module uses CSPs passed in by the calling application on the stack. The module does not store any CSP persistently (beyond the lifetime of an API call).

2.6.6 Zeroization

Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module.

Secret keys are provided to the Module by the calling application, and are destroyed when released by the appropriate API function calls. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the module defined API. The operating system protects memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently.

2.7 Electromagnetic Interference / Electromagnetic Compatibility

This section is not applicable.

2.8 Self Tests

Cryptographic self-tests are performed by the module on invocation of Initialize or Self-test, as well as when the module is operating in the FIPS-Approved mode. It should be noted that the output is inhibited while the module is in an error state. The following sections list the self-tests performed by the module, their expected error status, and any error resolutions.

2.8.1 Power Up Self Tests

The module performs the following tests upon power up:

- Software integrity check (HMAC SHA-1 Integrity Test)
- Known Answer Tests (KATs)
    - o AES Encryption KAT in ECB mode with 128-bit key
    - o AES Decryption KAT in ECB mode with 128-bit key
    - o HMAC SHA-1 KAT
    - o HMAC-SHA-256 KAT

The SHA algorthims are tested as part of the HMAC self tests.

The FIPS_mode_set()function performs all power-up self-tests listed above with no operator intervention required, returning a “1” if all power-up self-tests succeed, and a “0” otherwise. If any component of the power-up self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to FIPS_mode_set() succeeds.

2.9 Design Assurance

Configuration management for the module is provided by [CMS] which uniquely identifies each configuration item and the version of each configuration item.

Documentation version control is performed manually by updating the document date as well as the major and minor version numbers in order to uniquely identify each version of a document.

2.10 Mitigation of Other Attacks

The module does not claim to mitigate any attacks outside the requirements of FIPS 140-2.

3 Secure Operation

3.1 Configuration and Initialization

When installed, configured and initialized following these instructions the module provides access to FIPS Approved algorithms and security functions. To initialize the module the FIPS_mode_set () function is invoked. During initialization, the Power-On Self Test described in Section 2.8 Self Tests are run. If any component of the self tests fails, subsequent invocation of any cryptographic function calls will fail. FIPS_mode_set () initializes the module (FIPS_mode flag is TRUE) only if all tests are successful.

3.2 Crypto Officer Guidance

The installation of the module is performed by the Crypto Officer role.

The module is installed as follows:

- Copy fipscanister.o to the target machine. In order to maintain security of the module's operation, the Crypto Officer shall verify that the module is installed and executed in a physically secure location.
- Link fipscanister.o with the application code by running the corresponding make script.
- During application execution, call FIPS_mode_set () function and verify the return code is successful.
- The Cord3 cryptographic module is not available for public distribution, such as through a public web site. The Cord3 cryptographic module when compiled and linked, is only available from the Cord3 controlled source code repository for use by Cord3 products.

3.3 User Guidance

The module must be successfully configured and initialized to ensure proper operation. To initialize the module, invoke FIPS_mode_set() function which returns 1 for success and 0 for failure.

The calling application is responsible to interpret and handle the return code from the module.

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

In accordance to NIST guidance, operators are responsible for insuring that a single Triple-DES key shall not be used to encrypt more than 216 64-bit data blocks.
Historically, for FIPS 140-2 validated software cryptographic module on a server to meet the single user requirement of Security Level 1, the server has to be configured so that only one user at a time could access the server. The application that makes calls to the modules is the single user of the modules, even when the application is serving multiple clients

Please also note the guidance in Section 6.1 of NIST's publication Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program.

When a crypto module is implemented in a server environment, the server application is the user of the cryptographic module. The server application makes the calls to the cryptographic module.

Therefore, the server application is the single user of the cryptographic module, even when the server application is serving multiple clients.
When in FIPS mode, the operator must restrict their use of this module to only those cryptographic algorithms and functions that have been tested and certified in order to maintain FIPS compliance.

Appendix A Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CA	Certificate Authority
CBC	Cipher Block Chaining
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
ECC	Elliptic Curve Cryptography
EFP	Environmental Failure Protection
EMI/EMC	Electromagnetic Interference / Electromagnetic Compatibility
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards
HMAC	(Keyed-) Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
NVM	Non-Volatile Memory
PAA	Processor Algorithm Acceleration
ROM	Read Only Memory
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
TDES	Triple Data Encryption Standard
USB	Universal Serial Bus

Table 6 - Acronym Definitions

Appendix B Compilers

The specific compiler used to generate the Module for the Operational Environment is provided here. Note this does not imply that use of the Module is restricted to only the listed compiler version, only that the use of other versions has not been confirmed to produce a correct result.

#	Operational Environment	Compiler
1	Centos 7 64 bit	Gcc 4.8.5

Table 7 Compilers