FIPS 140-2 Security Policy

BlackBerry Linux Kernel Cryptographic Module, Version 1.0

Document version 1.1

BlackBerry Security Certifications, BlackBerry

Table of Contents

List of Figures

List of Tables

Introduction

BlackBerry® is the leading wireless solution that allows users to stay connected to a full suite of applications, including email, phone, enterprise applications, the Internet, Short Message Service (SMS), and organizer information. BlackBerry is a totally integrated package that includes innovative software, advanced BlackBerry wireless devices and wireless network service, providing a seamless solution. The BlackBerry® Enterprise Service 12 architecture is shown in the following figure.

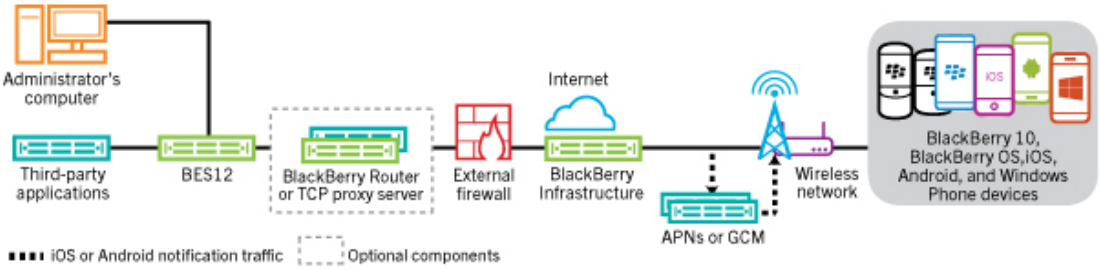


Figure 1. BlackBerry Enterprise Service 12 architecture

BlackBerry® smartphones are built on industry-leading wireless technology and, combined with BlackBerry Enterprise Service, provide users with an industry leading, end to end security solution. With the use of BlackBerry Enterprise Service 12, you can manage BlackBerry smartphones, as well as iOS® devices, Android™ devices, and Windows phones® all from a unified interface.

BlackBerry 10 smartphones contain the BlackBerry OS Cryptographic Library, a software module that provides the cryptographic functionality required for basic operation of the device. The BlackBerry Linux Kernel Cryptographic Module expands the secure capabilities and features BlackBerry is known for, to devices running operating systems other than the BlackBerry OS.

The BlackBerry Linux Kernel Cryptographic Module, hereafter referred to as the cryptographic module or module, provides the following cryptographic services:

- Data encryption and decryption
- Message digest and authentication code generation
- Random data generation

More information on the BlackBerry solution is available from http://ca.blackberry.com/.

The BlackBerry Linux Kernel Cryptographic Module meets the requirements applicable to FIPS 140-2 Security Level 1 as shown in Table 1.

Table 1. Summary of achieved security levels per FIPS 140-2 section

Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Cryptographic Module Security Policy	1

1 Cryptographic module specification

The BlackBerry Linux Kernel Cryptographic Module is a multiple-chip, stand-alone software cryptographic module in the form of an object that operates with the following components:

- Commercially available general-purpose computer hardware
- Commercially available OS that runs on the computer hardware

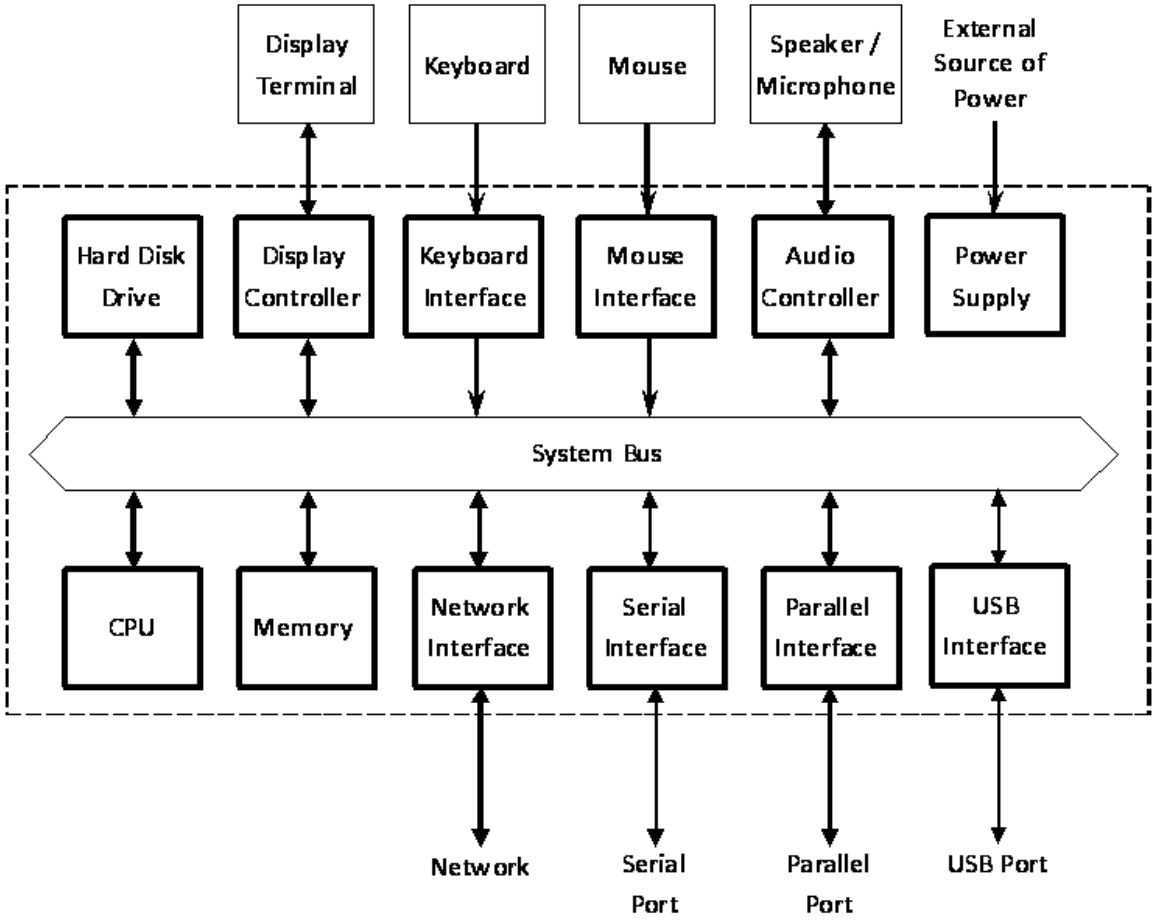
1.1 Physical specifications

The general, computer hardware component consists of the following devices:

- CPU (microprocessor)
- Working memory located on the RAM and contains the following spaces:
    - Input/Output buffer
    - Plaintext/ciphertext buffer
    - Control buffer
    - Program memory is also located on the RAM

  Note: Key storage is not deployed in this module.
- Hard disk (or disks), including flash memory
- Display controller, including the touch screen controller
- Keyboard interface
- Mouse interface, including the touch interface
- Audio controller
- Network interface
- Serial port
- Parallel port
- USB interface
- Power supply

Figure 2 illustrates the configuration of this component.



Key:

Cryptographic boundary

Flow of data, control input, and status output

Flow of control input

Flow of status output

Figure 2. Cryptographic module hardware block diagram

1.2 Computer hardware and OS

The combinations of computer hardware and OS include the following representative platforms:

1. Android 5.1 64-bit running on a BlackBerry PRIV
2. Centos 7 64-bit running on a Kontron NSN2U IP Network Server

The BlackBerry Linux Kernel Cryptographic Module is also suitable for any manufacturer's platform that has compatible processors, equivalent or larger system configurations, and compatible OS versions. The BlackBerry Linux Kernel Cryptographic Module will run on these platforms and OS versions while maintaining its compliance to the FIPS 140-2 Level 1 requirements.

1.3 Software specifications

The BlackBerry Linux Kernel Cryptographic Module is a software only security level 1 cryptographic module that provides general-purpose cryptographic services to the remainder of the Linux kernel. The binary of the module is in the Linux kernel's external module format.

The interface into the BlackBerry Linux Kernel Cryptographic Module is through application programming interface (API) function calls. These function calls provide the interface to the cryptographic services, for which the parameters and return codes provide the control input and status output as shown in Figure 3.

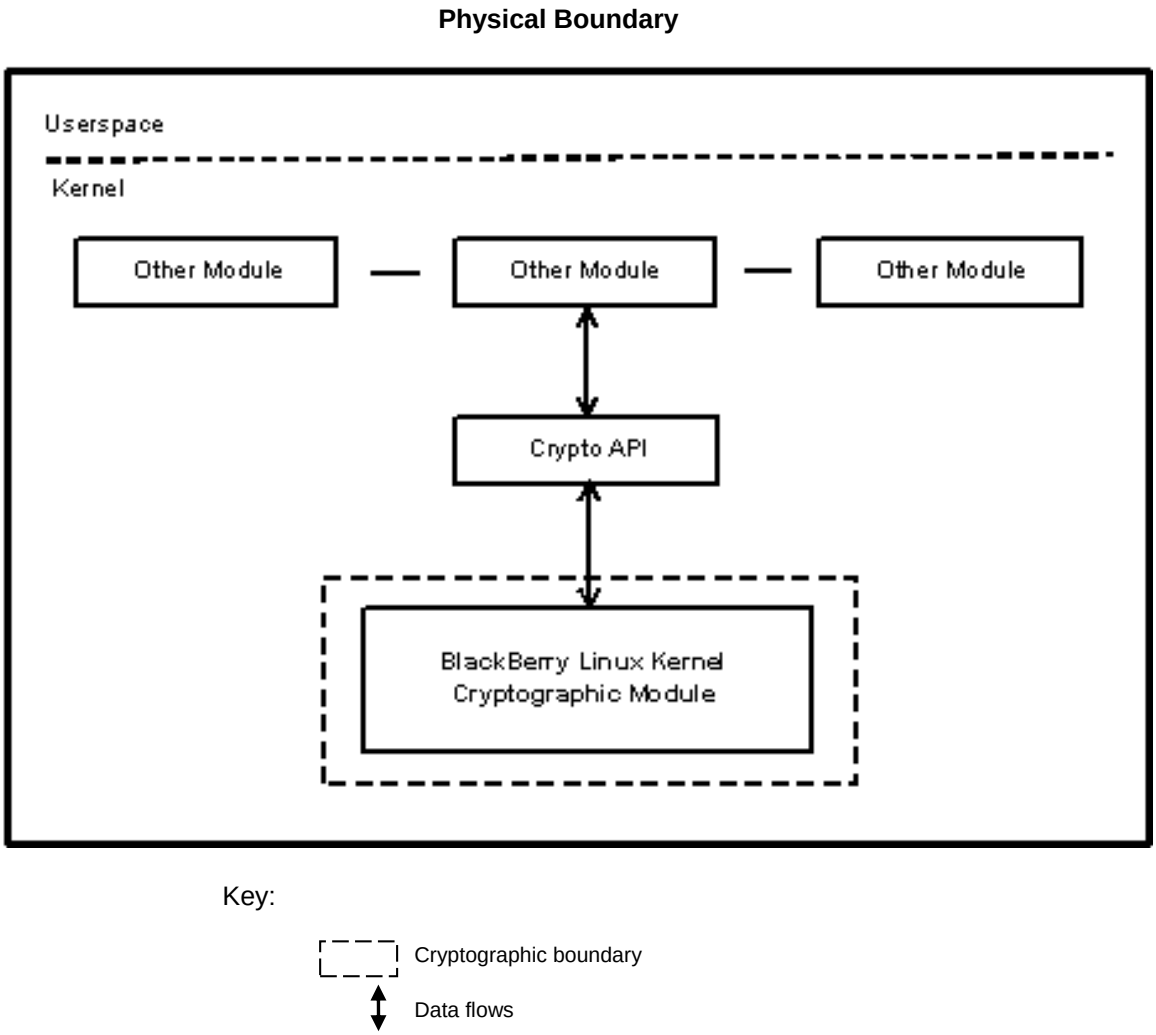


Figure 3: Cryptographic module software block diagram

2 Cryptographic module ports and interfaces

The cryptographic module ports correspond to the physical ports of the BlackBerry device that is executing the module, and the module interfaces correspond to the module's logical interfaces. The following table describes the module ports and interfaces.

Table 2. Implementation of FIPS 140-2 interfaces

FIPS 140-2 interface	Module ports
Data Input	API input parameters
Data Output	API output parameters
Control Input	API calls, module parameters
Status Output	API return code, console, Kernel log ring buffer
Power Input	API Initialization function
Maintenance	Not supported

The BlackBerry Linux Kernel Cryptographic Module is a normal Kernel external module aggregating a collection of cryptographic algorithms. The BlackBerry Linux Kernel Cryptographic Module does not make any changes to the API of using these algorithms.

For API usage, please refer to the publicly available Kernel Cryptographic API documents or the Linux kernel document under the kernel source tree (Documentation/crypto/api-intro.txt). Also many examples are available in the regression test module (tcrypt.c) in the kernel source.

Besides Kernel Cryptographic API, the BlackBerry Linux Kernel Cryptographic Module provides extra functions to do on-demand self-tests and to return module status, which are described in the Crypto Officer and User Guide in Appendix A.

3 Roles, services, and authentication

3.1 Roles and services

The module supports User and Crypto Officer roles. The module does not support a maintenance role. The module does not support multiple or concurrent operators and is intended for use by a single operator; thus it always operates in single-user mode.

Table 3. Roles and services

Services	Crypto Officer	User
Initialization services		
Initialization	X	
Deinitialization	X	
Self-tests	X	X
Show status	X	X
Key and CSP zeroization	X	X
Symmetric ciphers (AES, Triple-DES)		
Encrypt	X	X
Decrypt	X	X
Hash algorithms and message authentication (SHA, HMAC)		
Hashing	X	X
Message authentication	X	X
Random number generation (DRBG)		
Instantiation	X	
Seeding	X	X
Request	X	X

To operate the module securely, the Crypto Officer and User are responsible for confining those methods that have been FIPS 140-2 Approved. Thus, in the Approved mode of operation, all roles shall confine themselves to calling FIPS Approved algorithms, as shown in Table 4.

3.2 Security function

The BlackBerry Linux Kernel Cryptographic Module supports many cryptographic algorithms. Table 4 shows the set of cryptographic algorithms supported by the BlackBerry Linux Kernel Cryptographic Module.

Table 4. Supported cryptographic algorithms

	Algorithm	FIPS Approved or Allowed	Certificate number
Block ciphers	DES		
	Triple-DES (ECB, CBC) [NIST SP 800-67]	X	#1953
	AES (ECB, CBC, CTR, CCM, XTS) [FIPS 197]	X	#3464
	AES(GCM)		#3464
	AES(LRW)		
Hash functions	SHA-1 [FIPS 180-4]	X	#2859
	SHA-224 [FIPS 180-4]	X	#2859
	SHA-256 [FIPS 180-4]	X	#2859
	SHA-384 [FIPS 180-4]	X	#2859
	SHA-512 [FIPS 180-4]	X	#2859
Message authentication	HMAC-SHA-1 [FIPS 198-1]	X	#2209
	HMAC-SHA-224 [FIPS 198-1]	X	#2209
	HMAC-SHA-256 [FIPS 198-1]	X	#2209
	HMAC-SHA-384 [FIPS 198-1]	X	#2209
	HMAC-SHA-512 [FIPS 198-1]	X	#2209
Random	DBRG [NIST SP 800-90A]	X	#850

	Algorithm	FIPS Approved or Allowed	Certificate number
Number Generation	ANSI X9.31 RNG [ANSI X9.31]		#1383

On hardware platform Centos 7 64-bit running on a Kontron NSN2U IP Network Server, AES (ECB, CTR, CCM, and XTS modes) is tested with and without AES-NI instructions. AES (LRW) is only available with AES-NI instructions.

The 3-key Triple-DES, AES (ECB, CTR, CCM, and XTS modes), SHS (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512), HMAC-SHS (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA256, HMAC-SHA-384, and HMAC-SHA-512), DRBG (NIST SP 800-90), have been validated to comply with FIPS. In order to operate the module in compliance with FIPS, only these FIPS Approved or allowed algorithms should be used.

The AES (LRW, GCM), ANSI X9.31 Random Number Generation and DES are supported as non-FIPS Approved algorithms. In order to operate the module in compliance with FIPS, these algorithms should not be used.

Table 5 summarizes the keys and CSPs used in the FIPS mode.

Table 5. Key and CSP, key size, security strength, and access

Algorithm	Key and CSP	Key size	Security strength	Access
AES	Key	128, 192, 256 bits	128, 192, 256 bits	Read, Use
TDES	Key	168 bits	112 bits	Read, Use
HMAC	Key	160 to 512 bits	160 to 512 bits	Use
DRBG	seed	192-384 bits	128-256 bits	Use

Note:

- HMAC-SHA-1 shall have a key size of at least 112 bits.

3.3 Operator authentication

The BlackBerry Linux Kernel Cryptographic Module does not deploy an authentication mechanism. The operator implicitly selects the Crypto Officer and User roles.

4 Finite State Model

The Finite State Model contains the following states:

- Unloaded/Uninitialized
- Initialized
- Self-Test
- Idle
- Crypto Officer/User
- Error

The following list provides the important features of the state transitions:

1. When the Crypto Officer installs the module, the module is in the Unloaded /Uninitialized state.
2. When the initialization command is applied to the module, the module is loaded into memory and transitions to the Initialized state. Then, the module transitions to the Self-Test state and automatically runs the power-up tests. On success, the module enters the Idle state; on failure, the module enters the Error state and the module is disabled. From the Error state, the Crypto Officer might need to reinstall the module to attempt correction.
3. From the Idle state, which is entered only if the self-test has succeeded, the module can transition to the Crypto Officer/User state when an API function is called.
4. When the API function has completed successfully, the state transitions back to the Idle state.
5. If the conditional test fails, the state transitions to the Error state and the module is disabled.
6. When the on-demand self-test is executed, the module enters the Self-Test state. On success, the module enters the Idle state; on failure, the module enters the Error state and the module is disabled.
7. When the unload/de-initialization command is executed, the module returns to the Unloaded /Uninitialized state.
8. In the disabled state, the module can no longer be used.

5 Physical security

Physical security is not applicable to this software module at Level 1 Security.

6 Operational environment

The BlackBerry Linux Kernel Cryptographic Module runs on a single-user operational environment where each user application runs in a virtually separated, independent space.

Note: Modern operating systems, such as Linux and Android provide such operational environments.

7 Cryptographic key management

The BlackBerry Linux Kernel Cryptographic Module does not provide any key generation services. Keys must be established externally and passed into the module via API. The operating system protects unauthorized access to the keys and CSPs in the address space of the module process.

7.1 Random number generation

The BlackBerry Linux Kernel Cryptographic Module provides a FIPS Approved random number generator, DRBG (Hash, HMAC and CTR). The user must provide the seed, and must ensure that the seed is consistent with FIPS 140-2 requirement

7.2 Key generation

The cryptographic module does not perform key generation. The module implements a compliant NIST SP 800-90A DRBG which is exclusively used to generate random bits which are used by the calling application.

7.3 Key entry and output

The BlackBerry Linux Kernel Cryptographic Module does not support manual key entry and key output. Keys and other CSPs can only be exchanged between the module and the calling application via API parameters.

7.4 Key storage

The BlackBerry Linux Kernel Cryptographic Module does not provide persistent key storage.

7.5 Key zeroization

Kernel function kzfree() is used to de-allocate internal and intermediate generated CSPs in memory. This function guarantees zeroization occurs during de-allocation. All keys and CSPs can be zeroized by powering off the module and performing a system reset by the operational environment.

8 Self-tests

8.1 Power-up tests

Self-tests are initiated automatically by the module at start-up. The following tests are applied.

Table 6. Module self-tests

Test	Description
Known Answer Tests (KATs)	• AES(CBC, ECB, CTR, XTS, CCM) encryption and decryption • Triple-DES(CBC, ECB) encryption and decryption • HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 • SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 • ANSI X9.31 RNG • DRBG(CTR, HASH, HMAC)
Software integrity test	The software integrity test deploys SHA-256-HMAC to verify the integrity of the module

8.2 On-demand self-tests

The Crypto Officer or User can invoke on-demand self-tests by invoking a function, which is described in *Appendix C Crypto Officer and User Guide* in this document.

8.3 Conditional tests

The Continuous Test is executed on all RNG (DRBG and ANSI X9.31 RNG) generated data, examining each requested random generation for repetition. This ensures that the DRBG is not stuck at any constant value.

8.4 Critical Function tests

For DRBG (CTR, HASH, and HMAC), the module implements following critical function tests:

- SP 800-90 DRBG Instantiate Health Test
- SP 800-90 DRBG Generate Health Test
- SP 800-90 DRBG Reseed Health Test
- SP 800-90 DRBG Uninstantiate Health Test

9 Design assurance

9.1 Configuration management

A configuration management system for the cryptographic module is employed and has been described in documentation submitted to the testing laboratory. The module uses Subversion (SVN) to track the configurations.

9.2 Delivery and operation

To review the steps necessary for the secure installation and initialization of the cryptographic module, see *Appendix C – Crypto Officer and User Guide section C.1*.

9.3 Development

Detailed design information and procedures have been described in documentation that was submitted to the testing laboratory. The source code is fully annotated with comments, and it was also submitted to the testing laboratory.

9.4 Guidance documents

The *Crypto Officer Guide and User Guide,* provided as Appendix C, outlines the operations for the Crypto Officer and User to ensure the security of the module.

10 Mitigation of other attacks

No other attacks are mitigated

Appendix A Acronyms

Introduction

This appendix lists the acronyms used in this document.

Acronyms

Acronym	Full term
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
ARC	Alleged Rivest's Cipher
CBC	cipher block chaining
CCM	Counter with CBC-MAC
CFB	cipher feedback
CMAC	Cipher-based MAC
CSP	critical security parameter
CTR	counter
CVS	Concurrent Versioning System
DES	Data Encryption Standard
DRBG	deterministic random bit generator
DSA	Digital Signature Algorithm
ECB	electronic codebook
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
HMAC	Hash-based Message Authentication code
IEEE	Institute of Electrical and Electronics Engineers
KAT	known answer test

Acronym	Full term
MD	Message Digest Algorithm
NIST	National Institute of Standards and Technology
NDRNG	Non-Deterministic Random Number Generator
OAEP	Optimal Asymmetric Encryption Padding
OFB	output feedback
PIM	personal information management
PIN	personal identification number
RFC	Recursive Flow Classification
RNG	random number generator
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Service
SMS	Short Message Service
SVN	Subversion
TDES	Triple Data Encryption Standard
USB	Universal Serial Bus

Appendix B References

Introduction

This appendix lists the references that were used for this project.

References

1. *NIST Security Requirements For Cryptographic Modules, FIPS PUB 140-2, December 3, 2002*
2. *NIST Security Requirements For Cryptographic Modules, Annex A: Approved Security Functions for FIPS PUB 140-2, Draft, October 8, 2014.*
3. *NIST Security Requirements For Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2, Draft, August 12, 2011*
4. *NIST Security Requirements For Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2, Draft, February 16, 2012.*
5. *NIST Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2, Draft, October 8, 2014.*
6. *NIST Security Requirements For Cryptographic Modules Derived Test Requirements for FIPS PUB 140-2, Draft, January 4, 2011.*
7. *NIST Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, August 7, 2015.*
8. *NIST Frequently Asked Questions for the Cryptographic Module Validation Program, June 5, 2014.*

Appendix C Crypto Office and User Guide

C.1 Installation

In order to carry out a secure installation of the BlackBerry Linux Kernel Cryptographic Module, the Crypto Officer must follow the procedure described in this section.

C.1.1 Installing the cryptographic module

The Crypto Officer is responsible for the installation of the BlackBerry Linux Kernel Cryptographic Module. Only the Crypto Officer is allowed to install the product.

Note: Place the object in an appropriate location on the computer hardware.

C.1.2 Uninstalling the cryptographic module

Remove the object from the computer hardware.

C.2 Commands

C.2.1 Load/Initialization

insmod fipsm.ko

This root command loads the kernel module into memory and triggers a series of initialization and self-tests on the module. These tests examine the integrity of the object, and the correct operation of the cryptographic algorithms. If these tests are successful, the module will be enabled, otherwise, the module will output an error message saying the module loading is failed and the module will be unloaded..

C.2.2 Unload/Deinitialization

rmmod fipsm

This root command will de-initialize the module and unload it from memory. In some cases (platform dependent), the module may be used by other kernel modules and rmmod command could fail. At this time, a system reboot is required to unload/de-initialize the module.

C.2.3 Self-tests

int fipsmod_selftest(void)

This function runs a series of self-tests, and return zero if the tests are successful. These tests examine the integrity of the object, and the correct operation of the cryptographic algorithms. If these tests fail, the function will return a non-zero error code and the module will be disabled. Meanwhile, a message “fipsm: module disabled” can be seen from kernel log dmesg. Section C.3 of this document describes how to recover from the disabled state..

C.2.4 Show Status

int fipsmod_get_state(void)

This function will return the current state of the module.

The state value could be:

```
#define FIPSMOD_STATE_ENABLED      1
#define FIPSMOD_STATE_DISABLED     2
```

C.2.5 Module logs

insmod fipsm.ko v=1

The BlackBerry Linux Kernel Cryptographic Module can write logs into Kernel’s log ring buffer. Users can use dmesg to view the logs. The kernel module has a module parameter v (0-5) to control verbose level of the logs. If v is not specified or v=0, log will be turned off.

C.3 When the cryptographic module is disabled

When BlackBerry Linux Kernel Cryptographic Module becomes disabled, attempt to bring the module back to the Initialized state by unloading and reloading the module. If the initialization is successful, the module is recovered. If this recovery attempt fails, it indicates a fatal error. Please contact BlackBerry Support immediately.

Document and contact information

Version	Date	Author	Reason for revision
1.0	June 23, 2016	Randy Eyamie	Original release
1.1	June 30, 2016	Randy Eyamie	Minor edits

Contact	Corporate office
Security Certifications Team certifications@blackberry.com (519) 888-7465 ext. 72921	BlackBerry B 2200 University Ave. E Waterloo, ON, Canada N2K 0A7 www.blackberry.com