

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

---

# **FIPS 140-2 Non-Proprietary Security Policy**

---

**BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9**

**Document version 1.14**

**BlackBerry Security Certifications, BlackBerry**

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## Table of Contents

|  |           |
|--|-----------|
| <b>TABLE OF CONTENTS</b> .....                           | <b>2</b>  |
| <b>LIST OF FIGURES</b> .....                             | <b>4</b>  |
| <b>LIST OF TABLES</b> .....                              | <b>5</b>  |
| <b>INTRODUCTION</b> .....                                | <b>6</b>  |
| <b>1 CRYPTOGRAPHIC MODULE SPECIFICATION</b> .....        | <b>8</b>  |
| 1.1 PHYSICAL SPECIFICATIONS .....                        | 8         |
| 1.2 COMPUTER HARDWARE, OS, AND JVM .....                 | 10        |
| 1.3 SOFTWARE SPECIFICATIONS .....                        | 11        |
| <b>2 CRYPTOGRAPHIC MODULE PORTS AND INTERFACES</b> ..... | <b>12</b> |
| <b>3 ROLES, SERVICES, AND AUTHENTICATION</b> .....       | <b>13</b> |
| 3.1 ROLES AND SERVICES .....                             | 13        |
| 3.2 SECURITY FUNCTION .....                              | 14        |
| 3.3 OPERATOR AUTHENTICATION .....                        | 18        |
| <b>4 FINITE STATE MODEL</b> .....                        | <b>19</b> |
| <b>5 PHYSICAL SECURITY</b> .....                         | <b>20</b> |
| <b>6 OPERATIONAL ENVIRONMENT</b> .....                   | <b>21</b> |
| <b>7 CRYPTOGRAPHIC KEY MANAGEMENT</b> .....              | <b>22</b> |
| 7.1 KEY GENERATION .....                                 | 22        |
| 7.2 KEY ESTABLISHMENT .....                              | 22        |
| 7.3 KEY ENTRY AND OUTPUT .....                           | 22        |
| 7.4 KEY STORAGE .....                                    | 22        |
| 7.5 KEY ZEROIZATION .....                                | 23        |
| <b>8 SELF-TESTS</b> .....                                | <b>24</b> |
| 8.1 POWER-UP TESTS .....                                 | 24        |
| 8.2 ON-DEMAND SELF-TESTS .....                           | 24        |
| 8.3 CONDITIONAL TESTS .....                              | 24        |
| 8.4 FAILURE OF SELF-TESTS .....                          | 24        |
| <b>9 DESIGN ASSURANCE</b> .....                          | <b>25</b> |
| 9.1 CONFIGURATION MANAGEMENT .....                       | 25        |
| 9.2 DELIVERY AND OPERATION .....                         | 25        |
| 9.3 DEVELOPMENT .....                                    | 25        |
| 9.4 GUIDANCE DOCUMENTS .....                             | 25        |
| <b>10 MITIGATION OF OTHER ATTACKS</b> .....              | <b>26</b> |
| 10.1 TIMING ATTACK ON RSA .....                          | 26        |

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

|   |   |           |
|---|---|-----------|
| 10.2  | ATTACK ON BIASED PRIVATE KEY OF DSA ..... | 26        |
| <b>DOCUMENT AND CONTACT INFORMATION .....</b> |   | <b>32</b> |

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## List of Figures

|  |    |
|--|----|
| Figure 1. BlackBerry Enterprise Service 10 architecture..... | 6  |
| Figure 2. Cryptographic module hardware block diagram.....   | 9  |
| Figure 3: Cryptographic module software block diagram.....   | 11 |

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

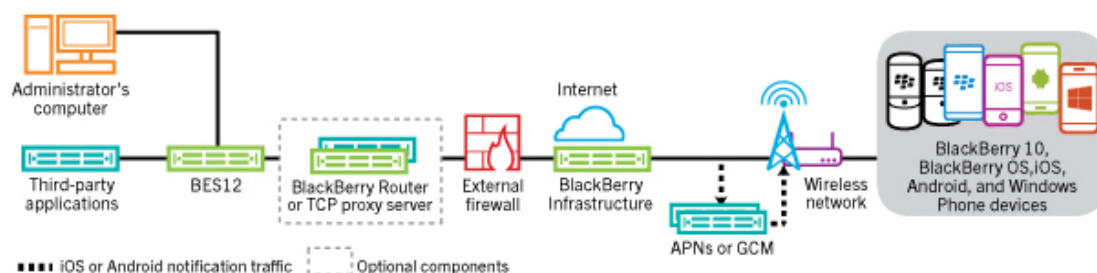
**List of Tables**

|   |    |
|---|----|
| Table 1. Summary of achieved security levels per FIPS 140-2 section ..... | 7  |
| Table 2. Implementation of FIPS 140-2 interfaces .....                    | 12 |
| Table 3. Roles and services .....   | 13 |
| Table 4. Supported cryptographic algorithms.....                          | 14 |
| Table 5. Key and CSP, key size, security strength, and access .....       | 17 |
| Table 6. Module self-tests .....  | 24 |

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 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 Cryptographic Java Module expands the secure capabilities and features BlackBerry is known for, to devices running operating systems other than the BlackBerry OS.

The BlackBerry Cryptographic Java Module, hereafter referred to as cryptographic module or module, is a software module that provides the following cryptographic services to the BlackBerry Enterprise Service 12 and other BlackBerry device management components:

- Data encryption and decryption
- Message digest and authentication code generation
- Random data generation
- Elliptic curve key pair generation
- Elliptic curve digital signature generation and verification
- Elliptic curve key agreement

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

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

The BlackBerry Cryptographic Java 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               | 1     |
| Cryptographic Module Security Policy      | 1     |

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 1 Cryptographic module specification

The BlackBerry Cryptographic Java Module is a multiple-chip, stand-alone software cryptographic module that operates with the following components:

- Commercially available general-purpose computer hardware
- Commercially available Operating System (OS) that runs on the computer hardware
- A commercially available Java Virtual Machine (JVM) that runs on the computer hardware and OS

### 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

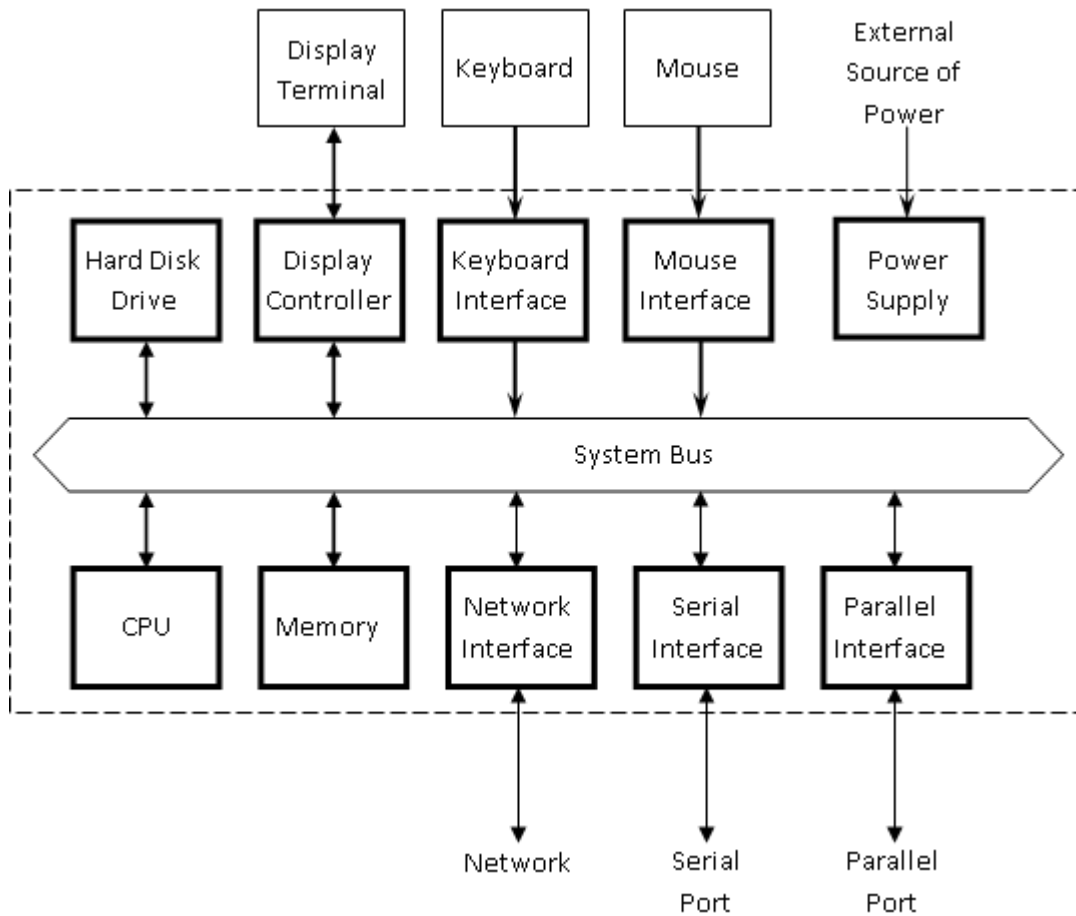
**Note:** Key storage is not deployed in this module.

- Program memory is also located on the RAM
- Hard disk (or disks), including flash memory
- Display controller, including the touch screen controller
- Keyboard interface
- Mouse interface, including the trackball interface
- Audio controller
- Network interface
- Serial port
- Parallel port
- USB interface
- Power supply

Figure 2 illustrates the configuration of this component.



BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9



Key:

- ▭▭▭ Physical 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**

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 1.2 Computer hardware, OS, and JVM

The BlackBerry Cryptographic Java Module versions 2.8 and 2.8.7 have been tested on the following representative combinations of computer hardware and OS, running the Java Runtime Environment (JRE) 1.5.0 and 1.6.0 by Sun Microsystems:

1. Solaris 10, 32-bit SPARC (Binary compatible to Solaris 9)
2. Solaris 10, 64-bit SPARC (Binary compatible to Solaris 9)
3. Red Hat Linux AS 5.5, 32-bit x86 (Binary compatible to AS 2.1/3.0/4.0/5.0)
4. Red Hat Linux AS 5.5, 64-bit x86 (Binary compatible to AS 4.0/5.0)
5. Windows Vista, 32-bit x86 (Binary compatible to Windows 98/2000/2003/XP)
6. Windows Vista, 64-bit x86 (Binary compatible to Windows 64-bit XP).
7. Windows 2008 Server, 64-bit x86

The BlackBerry Cryptographic Java Module versions 2.8.8 and 2.9 have been tested on the following representative combinations of computer hardware and OS, running the Java Runtime Environment (JRE) 1.8.0 by Oracle:

1. CentOS Linux 7.0 64-bit x86

The module will run on the JREs 1.3.1, and 1.4.2, and on various hardware and OS such as,

1. Any other Solaris Platforms,
2. Any other Linux Platforms,
3. Any other Windows Platforms,
4. AIX Platforms, and
5. HP-UX Platforms,

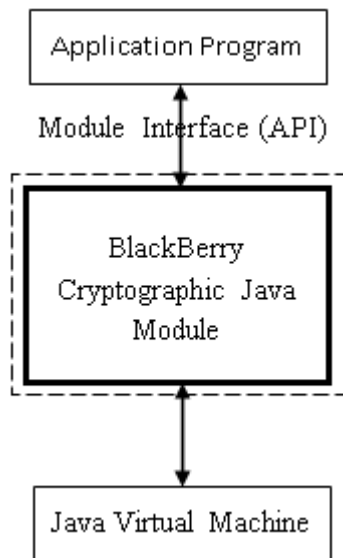
while maintaining its compliance to the FIPS 140-2 Level 1 requirements. Thus, this validation is applicable to these JREs and platforms as well.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9



### 1.3 Software specifications

The BlackBerry Cryptographic Java Module provides services to the Java computer language users in the form of a Java archive (JAR). The same binary is used for all identified computer hardware and OS because the JVM underneath the BlackBerry Cryptographic Java Module will absorb the differences of the computer hardware and OS.

The interface into the BlackBerry Cryptographic Java Module is through Application Programmer's Interface (API) method calls. These method calls provide the interface to the cryptographic services, for which the parameters and return codes provide the control input and status output (see Figure 3).



Key:

-  Cryptographic boundary
-  Data flows

**Figure 3: Cryptographic module software block diagram**

## 2 Cryptographic module ports and interfaces

The cryptographic module ports correspond to the physical ports of the GPC 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            | Module interfaces                        |
|----------------------|-------------------------|--|
| Data Input           | API                     | Ethernet Port                            |
| Data Output          | API                     | Ethernet Port                            |
| Control Input        | API                     | Keyboard and Mouse                       |
| Status Output        | Return Code             | Display                                  |
| Power Input          | Initialization Function | The Power Supply is the power interface. |
| Maintenance          | Not supported           | Not supported                            |

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 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 a single-user mode.

**Table 3. Roles and services**

| Service* <sup>1</sup>   | Crypto Officer | User |
|---|----------------|------|
| <b>Initialization, etc.</b>                                   |                |      |
| Initialization  | x              | x    |
| Deinitialization  | x              | x    |
| Self-tests  | x              | x    |
| Show status   | x              | x    |
| <b>Symmetric Ciphers (AES and TRIPLE-DES)</b>                 |                |      |
| Key generation (Triple-DES only)                              | x              | x    |
| Encrypt   | x              | x    |
| Decrypt   | x              | x    |
| Key zeroization   | x              | x    |
| <b>Hash Algorithms and Message Authentication (SHA, HMAC)</b> |                |      |
| Hashing   | x              | x    |
| Message authentication  | x              | x    |
| <b>Random Bit Generation</b>                                  |                |      |
| Instantiation   | x              | x    |
| Request   | x              | x    |
| CSP/key zeroization   | x              | x    |

<sup>1</sup> These services are non-Approved when using one of the non-Approved algorithms as specified in table 4

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

| Service*1   | Crypto Officer | User |
|---|----------------|------|
| <b>Digital Signature (DSA, ECDSA, RSA)</b>                                  |                |      |
| Key pair generation   | x              | x    |
| Sign  | x              | x    |
| Verify  | x              | x    |
| Key Zeroization   | x              | x    |
| <b>Key Agreement (Diffie-Hellman, Elliptic Curve Diffie-Hellman, ECMQV)</b> |                |      |
| Key pair generation   | x              | x    |
| Shared secret generation  | x              | x    |
| Key Zeroization   | x              | x    |
| <b>KeyWrapping (RSA)</b>  |                |      |
| Key pair generation   | x              | x    |
| Wrap  | x              | x    |
| Unwrap  | x              | x    |
| Key Zeroization   | 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 functions

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

**Table 4. Supported cryptographic algorithms**

| Type          | Algorithm  | FIPS approved or allowed | Certificate number    |
|---------------|--|--------------------------|-----------------------|
| Block Ciphers | DES (ECB, CBC, CFB64, OFB64)                       |                          |                       |
|               | TRIPLE-DES (TECB, TCBC, TCFB64, TOFB64) [SP800-67] | x                        | # 964, # 1954, # 2188 |

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

| Type                   | Algorithm  | FIPS approved or allowed | Certificate number     |
|------------------------|--|--------------------------|------------------------|
|                        | DESX (ECB, CBC, CFB64, OFB64)                                  |                          |                        |
|                        | AES (ECB, CBC, CFB128, OFB128, CTR, CCM, CMAC, GCM) [FIPS 197] | x                        | # 1411, # 3465, # 3988 |
|                        | ARC2 (ECB, CBC, CFB64, OFB64) [RFC 2268]                       |                          |                        |
| Stream Cipher          | ARC4   |                          |                        |
| Hash Functions         | SHA-1 [FIPS 180-4]   | x                        | # 1281, # 2860, # 3292 |
|                        | SHA-224 [FIPS 180-4]   | x                        | # 1281, # 2860, # 3292 |
|                        | SHA-256 [FIPS 180-4]   | x                        | # 1281, # 2860, # 3292 |
|                        | SHA-384 [FIPS 180-4]   | x                        | # 1281, # 2860, # 3292 |
|                        | SHA-512 [FIPS 180-4]   | x                        | # 1281, # 2860, # 3292 |
|                        | MD5 [RFC 1321]   |                          |                        |
|                        | MD2 [RFC 1115]   |                          |                        |
|                        | RIPEMD   |                          |                        |
| Message Authentication | HMAC-SHA-1 [FIPS 198-1]  | x                        | # 832, # 2210, # 2603  |
|                        | HMAC-SHA-224 [FIPS 198-1]                                      | x                        | # 832, # 2210, # 2603  |
|                        | HMAC-SHA-256 [FIPS 198-1]                                      | x                        | # 832, # 2210, # 2603  |
|                        | HMAC-SHA-384 [FIPS 198-1]                                      | x                        | # 832, # 2210, # 2603  |
|                        | HMAC-SHA-512 [FIPS 198-1]                                      | x                        | # 832, # 2210, # 2603  |
|                        | HMAC-MD5 [RFC 2104]  |                          |                        |

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

| Type  | Algorithm                                | FIPS approved or allowed         | Certificate number     |
|---|--|----------------------------------|------------------------|
| Random Bit Generation                           | ANSI X9.62 RNG [ANSI X9.62]              |                                  |                        |
|   | DRBG [NIST SP 800-90A Rev. 1]            | x                                | # 52, # 852, # 1180    |
|   | NDRBG (GenerateSeed())                   | x                                |                        |
| Digital Signature                               | DSA [FIPS 186-2]<br>DSA [FIPS 186-4]     | x                                | # 455<br># 978, # 1084 |
|   | ECDSA [FIPS 186-2]<br>ECDSA [FIPS 186-4] | x                                | # 179<br># 702, # 884  |
|   | ECDSA [ANSI X9.62]                       |                                  |                        |
|   | RSA PKCS1 v1.5 Signature[PKCS #1 v2.1]   | x                                | # 687, # 1776, # 2046  |
|   | RSA PSS [PKCS #1 v2.1]                   | x                                | # 687, # 1776, # 2046  |
|   | ECQV                                     |                                  |                        |
|   | Key Agreement                            | Diffie-Hellman [NIST SP 800-56A] | x                      |
| Elliptic Curve Diffie-Hellman [NIST SP 800-56A] |  | x                                | # 8, # 62, # 83        |
| ECMQV [NIST SP 800-56A]                         |  | x                                | # 8, # 62, # 83        |
| Key Wrapping                                    | RSA PKCS1 v1.5 Encryption [PKCS #1 v2.1] |                                  |                        |
|   | RSA OAEP [NIST SP 800-56B]               | x                                |                        |
|   | ECIES [ANSI X9.63]                       |                                  |                        |

The DES, DESX, AES CCM\* (CCM star) mode, random bit generation (ANSI X9.62), ARC2, ARC4, RIPEMD, MD4, MD2, and HMAC-MD5, ECQV, ECIES, RSA PKCS #1 v1.5 Encryption algorithm, and Diffie-Hellman with strength < 112 bits are supported as non FIPS Approved algorithms. In order to operate the module in a FIPS Approved mode of operation these algorithms must not be used. For versions 2.8, 2.8.7 and 2.8.8, AES GCM encryption shall not be performed in order to remain FIPS compliant. If the module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.



## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

**Note:** 2-Key Triple-DES decryption is permitted for legacy purposes. 2-Key Triple-DES encryption is considered a non FIPS Approved algorithm as of January 1<sup>st</sup>, 2016. Please consult NIST SP 800-131A for additional details on algorithm transitions.

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 to 256 bits   | 128 to 256 bits   | Use               |
| TRIPLE-DES       | Key                       | 192 bits  | 112 bits          | Create, Read, Use |
| HMAC             | Key                       | 160 to 512 bits   | 160-512 bits      | Use               |
| DRBG             | seed                      | 160-512 bits  | 160-256 bits      | Use               |
| DSA              | Key pair                  | 2048 to 15360 bits                                      | 112 to 256 bits   | Create, Read, Use |
| ECDSA            | Key pair                  | 224 to 521 bits   | 112 to 256 bits   | Create, Read, Use |
| RSA              | Key pair                  | 2048 to 15360 bits                                      | 112 to 256 bits   | Create, Read, Use |
| DH               | Static/ephemeral key pair | Public key 2048-15360 bits/<br>Private key 224-512 bits | 112 to 256 bits   | Create, Read, Use |
| ECDH             | Static/ephemeral key pair | 224 to 521 bits   | 112 to 256 bits   | Create, Read, Use |
| ECMQV            | Static/ephemeral key pair | 224 to 521 bits   | 112 to 256 bits   | Create, Read, Use |
| RSA key wrapping | Key pair                  | 2048 to 15360 bits                                      | 112 to 256 bits   | Create, Read, Use |

Note:

- Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength (public key between 2048 and 15360 bits, private key between 224 and 512 bits); non-compliant less than 112-bits of encryption strength (public key less than 2048 bits, private key less than 224 bits)).
- EC Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength; non-compliant less than 112-bits of encryption strength).
- ECMQV (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength; non-compliant less than 112-bits of encryption strength).

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

- RSA (key wrapping; key establishment methodology provides between 112 and 256 bits of encryption strength; non-compliant less than 112-bits of encryption strength).
- Digital signature generation that provides less than 112 bits of security (using RSA, DSA or ECDSA) is disallowed beginning January 1st, 2014.
- Digital signature generation using SHA-1 as its underlying hash function is disallowed beginning January 1st, 2014.
- HMAC-SHA-1 shall have a key size of at least 112 bits.
- In FIPS approved mode only the curves P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571 can be used.
- The BlackBerry Cryptographic Java Module supports the elliptic curves K-163, B-163, P-192, secp160r1, sect239k1 and wTLS5 that are not FIPS approved. They can be used with the ECDSA, ECDH, ECMQV and ECIES algorithms, but not in FIPS approved mode.
- In version 2.9, the operator shall use an Approved DRBG to generate the IV when using AES GCM

### **3.3 Operator authentication**

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

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 4 Finite State Model

The Finite State Model contains the following states:

- Installed/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 Installed/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. While in the Self-Test state, all data output through the data output interface is prohibited. 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 (continuous RNG test or Pair-wise consistency 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 de-initialization command is executed, the module returns to the Installed/Uninitialized state.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 5 Physical security

The BlackBerry device that executes this module is manufactured using industry standard integrated circuits and meets the FIPS 140-2 Level 1 physical security requirements.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 6 Operational environment

The BlackBerry Cryptographic Java 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 Unix, Linux, and Windows provide such operational environments.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 7 Cryptographic key management

The BlackBerry Cryptographic Java Module provides the underlying functions to support FIPS 140-2 Level 1 key management. The user will select FIPS Approved algorithms and will handle keys with appropriate care to build up a system that complies with FIPS 140-2. The Crypto Officer and User are responsible for selecting FIPS 140-2 validated algorithms. For more information, see Table 4.

### 7.1 Key generation

The BlackBerry Cryptographic Java Module provides FIPS 140-2 compliant key generation. The underlying random number generation uses a FIPS Approved method, DRBG.

The module also supports Dual\_EC DRBG; however, the use of Dual\_EC DRBG is non-approved for key generation. No keys generated using this version of the DRBG can be used to protect sensitive data in the Approved mode. Any random output in Approved mode using the DUAL\_EC DRBG is equivalent to plaintext.

### 7.2 Key establishment

The BlackBerry Cryptographic Java Module provides the following FIPS Approved or Allowed key establishment techniques [5]:

- Diffie Hellman (DH): The DH key agreement technique implementation supports modulus sizes from 512 bits to 15360 bits that provides between 56 and 256 bits of security strength, where 2048 bits and above must be used to provide a minimum of 112 bits of security in the FIPS mode.
- EC Diffie-Hellman (ECDH) & ECMQV : The ECDH and ECMQV key agreement technique implementations support elliptic curve sizes from 160 bits to 571 bits that provide between 80 and 256 bits of security strength, where 224 bits and above must be used to provide a minimum of 112 bits of security in the FIPS mode.
- RSA OAEP: The RSA OAEP key wrapping implementation supports modulus sizes from 512 bits to 15360 bits that provides between 56 and 256 bits of security, where 2048 bits and above must be used to provide minimum of 112 bits of security in the FIPS mode.

It is the responsibility of the calling application to make sure that the appropriate key establishment techniques are applied to the appropriate keys.

### 7.3 Key entry and output

Secret (security sensitive) keys must be imported to and exported from the cryptographic boundary in encrypted form using a FIPS Approved algorithm.

### 7.4 Key storage

The BlackBerry Cryptographic Java Module is a low-level cryptographic toolkit; therefore, it does not provide key storage.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## **7.5 Key zeroization**

The BlackBerry Cryptographic Java Module provides zeroizable interfaces which implement zeroization methods. Zeroization of all keys and CSPs are performed in the finalizing methods of the objects; JVM executes the finalizing methods every time it operates garbage collection.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 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) | <p>KATs are performed on Triple-DES encrypt, Triple-DES decrypt, AES encrypt, and AES decrypt, AES GCM, SHS (using HMAC-SHS), HMAC-SHS, DRBG Hash, DRBG HMAC, DRBG counter, RNG, RSA sign and RSA verify, and KDF. For DSA and ECDSA, a Pair-wise Consistency Test is used.</p> <p>For DH, ECDH, ECMQV, the underlying arithmetic implementations are tested using DSA and ECDSA tests.</p> |
| Software integrity test   | The software integrity test deploys ECDSA signature validation to verify the integrity of the module.   |
| DRBG Health tests         | DRBG Instantiate, DRBG Generate, DRBG Reseed, DRBG Un-instantiate   |

### 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 RNG test is executed on all RNG generated data, examining the first 160 bits of each requested random generator for repetition. This examination makes sure that the RNG is not stuck at any constant value. In addition, upon each generation of a DSA, ECDSA, or RSA key pair, the generated key pair is tested for its correctness by generating a signature and verifying the signature on a given message as a Pair-wise Consistency Test. Upon generation or reception of a DH, ECDH, or ECMQV key pair, the key pair is tested of their correctness by checking shared secret matching of two key agreement parties as a Pair-wise Consistency Test.

### 8.4 Failure of self-tests

Self-test failure places the cryptographic module in the Error state, wherein no cryptographic operations can be performed. The module is disabled. Additionally, the cryptographic module will throw a Java exception to the caller.



BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## **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 the Concurrent Versioning System (CVS) or Subversion (SVN) to track the configurations.

### **9.2 Delivery and operation**

Please refer to Section A.1 of *Crypto Officer And User Guide* in Appendix A to review the steps necessary for the secure installation and initialization of the cryptographic module.

### **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* outlines the operations for the Crypto Officer and User to ensure the security of the module.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 10 Mitigation of other attacks

The BlackBerry Cryptographic Java Module implements mitigation of the following attacks:

- Timing attack on RSA
- Attack on biased private key of DSA

### 10.1 Timing attack on RSA

When employing Montgomery computations, timing effects allow an attacker to tell when the base of exponentiation is near the secret modulus. This attack leaks information concerning the secret modulus.

In order to mitigate this attack, the bases of exponentiation are randomized by a novel technique that requires no inversion to remove (unlike other blinding methods, for example, see *BSAFE Crypto-C User Manual v4.2*).

Note: Remote timing attacks are practical. For more information, see *Remote Timing Attacks are Practical* [9].

### 10.2 Attack on biased private key of DSA

The standards for choosing ephemeral values in El-Gamal type signatures introduce a slight bias. Daniel Bleichenbacher presented the means to exploit these biases to ANSI.

In order to mitigate this attack, this bias in RNG is reduced to levels that are far below the Bleichenbacher attack threshold.

To mitigate this attack, NIST published Change Notice 1 of FIPS 186-2.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

# 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                      |
| DH      | Diffie-Hellman                                |
| DRBG    | deterministic random bit generator            |
| DSA     | Digital Signature Algorithm                   |
| EC      | Elliptic Curve                                |
| ECB     | electronic codebook                           |
| ECC     | Elliptic Curve Cryptography                   |
| ECDH    | Elliptic Curve Diffie-Hellman                 |
| ECDSA   | Elliptic Curve Digital Signature Algorithm    |
| ECIES   | Elliptic Curve Integrated Encryption Standard |
| ECMQV   | Elliptic Curve Menezes-Qu-Vanstone            |

## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

| Acronym    | Full term   |
|------------|---|
| ECNR       | Elliptic Curve Nyburg Rueppel                     |
| ECQV       | Elliptic Curve Qu-Vanstone                        |
| 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                                 |
| LCD        | liquid crystal display                            |
| LED        | light-emitting diode                              |
| MD         | Message Digest Algorithm                          |
| NIST       | National Institute of Standards and Technology    |
| OAEP       | Optimal Asymmetric Encryption Padding             |
| OFB        | output feedback                                   |
| PIM        | personal information management                   |
| PIN        | personal identification number                    |
| PKCS       | Public-Key Cryptography Standard                  |
| PSS        | Probabilistic Signature Scheme                    |
| pRNG       | pseudorandom number generator                     |
| 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  |
| TRIPLE-DES | Triple Data Encryption Standard                   |
| USB        | Universal Serial Bus                              |

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## 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, July 26, 2011*
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, July 26, 2011.*
5. *NIST Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2, Draft, July 26, 2011.*
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, July 15, 2011.*
8. *NIST Frequently Asked Questions for the Cryptographic Module Validation Program, December 4, 2007.*
9. David Brumley, Dan Boneh, "Remote Timing Attacks are Practical", *Stanford University*  
<http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf>

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

# Appendix C Crypto Office and User Guide

---

## C.1 Installation

In order to carry out a secure installation of the BlackBerry Cryptographic Java 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 Cryptographic Java Module. Only the Crypto Officer is allowed to install the product.

Note: Place the cryptographic module, EccpressoFIPS.jar in CLASSPATH or as in installed extension.

### C.1.2 Uninstalling the cryptographic module

Remove the jar file, EccpressoFIPS.jar, from the computer hardware.

## C.2 Commands

### C.2.1 Initialization

`FIPSManage.getInstance().activateFIPSMode()`

This method runs a series of Self-Tests on the module. These tests examine the integrity of the shared object, and the correct operation of the cryptographic algorithms. If these tests are successful, the module will be enabled.

For versions 2.8 and 2.8.7, you must invoke this method to enable the module.

For version 2.8.8 and 2.9, the Self-Tests are automatically run whenever a CryptoTransform or DRBG object is created. Therefore, there is no need to invoke this method to initialize the module explicitly.

If the module has been de-initialized by a previous call to `FIPSManage.getInstance().deactivateFIPSMode()`, it can be re-initialized by calling this method.

### C.2.2 Deinitialization

`FIPSManage.getInstance().deactivateFIPSMode()`

This method de-initializes the module.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

### **C.2.3 Self-tests**

`FIPSManage.getInstance().runSelfTests()`

This method runs a series of Self-Tests, and returns if the tests are successful, otherwise, an exception is thrown. These tests examine the integrity of the shared object, and the correct operation of the cryptographic algorithms. If these tests fail, the module will be disabled. Section C.3 of this document describes how to recover from the disabled state.

### **C.2.4 Show Status**

Status can be found by calling `FIPSManager.getInstance().isFIPSMode()` and `FIPSManager.getInstance().requestCryptoOperation()`. If both methods return true, the module is in the Idle state.

## **C.3 When the cryptographic module is disabled**

When BlackBerry Cryptographic Java Module becomes disabled, attempt to bring the module back to the Installed state by calling the deinitialization method, and then to initialize the module using the initialization method. If the initialization is successful, the module is recovered. If this attempt fails, uninstall the module and re-install it. If the module is initialized successfully by this re-installation, the recovery is successful. If this recovery attempt fails, it indicates a fatal error. Contact BlackBerry Support immediately.

### **C.1.2 FIPS Mode**

In order to operate in FIPS 140-2 mode, it is the user's responsibility to use FIPS Approved algorithms, as marked in Table 4. In particular, AES GCM encryption shall not be performed for versions 2.8, 2.8.7 and 2.8.8 of the module.

BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

## Document and contact information

| Version | Date              | Author       | Reason for revision                               |
|---------|-------------------|--------------|---|
| 1.0     | January 06, 2012  | Randy Eyamie | Document creation                                 |
| 1.1     | June 1, 2012      | Randy Eyamie | Updates based on Lab Comments                     |
| 1.2     | June 14, 2012     | Randy Eyamie | Added reference to version 2.8.7                  |
| 1.3     | May 29, 2015      | Randy Eyamie | Added reference to version 2.8.8                  |
| 1.4     | July 29, 2015     | Randy Eyamie | Updates based on Lab Comments                     |
| 1.5     | November 12, 2015 | Randy Eyamie | Updates based on Lab Comments                     |
| 1.6     | November 25, 2015 | Randy Eyamie | Updates based on Lab Comments                     |
| 1.7     | December 16, 2015 | Randy Eyamie | Updates based on Lab Comments                     |
| 1.8     | January 11, 2016  | Randy Eyamie | Updates required for NIST SP 800-131A transitions |
| 1.9     | June 13, 2016     | Randy Eyamie | Addition of version 2.9                           |
| 1.10    | June 14, 2016     | Randy Eyamie | Updates based on Lab Comments                     |
| 1.11    | September 1, 2016 | Randy Eyamie | Updates based on Lab Comments                     |
| 1.12    | September 1, 2016 | Randy Eyamie | Updates based on Lab Comments                     |
| 1.13    | October 4, 2016   | Randy Eyamie | Updates based on Lab Comments                     |
| 1.14    | October 5, 2016   | Randy Eyamie | Update to Table 4                                 |



## BlackBerry Cryptographic Java Module Versions 2.8, 2.8.7, 2.8.8 and 2.9

| Contact   | Corporate office  |
|---|---|
| Security Certifications Team<br><a href="mailto:certifications@blackberry.com">certifications@blackberry.com</a><br>(519) 888-7465 ext. 72921 | BlackBerry B<br>2200 University Ave. E<br>Waterloo, ON, Canada<br>N2K 0A7<br><a href="http://www.blackberry.com">www.blackberry.com</a> |