

FIPS 140-2 Security Policy

BlackBerry Cryptographic Kernel

Version 3.8.5.51



Document Version 2.6

Security Certifications Team

Research In Motion (RIM)

Document and Contact Information

| Version | Date | Description |
|---------|-----------------|---|
| 1.0 | 30 October 2008 | Document creation. |
| 1.1 | 5 December 2008 | Certificate numbers added in. |
| 2.0 | 15 January 2009 | Revised policy with minor modifications |
| 2.1 | 19 January 2009 | Added certificate numbers into table |
| 2.2 | 20 January 2009 | Revised policy with minor modifications |
| 2.3 | 1 May 2009 | Updated policy addressing CMVP comments |
| 2.4 | 13 May 2009 | Revised policy with minor modifications |
| 2.5 | 15 May 2009 | Revised policy with minor modifications |
| 2.6 | 25 May 2009 | Revised policy addressing comments from DOMUS |

| Contact | Corporate Office |
|--|--|
| Security Certifications Team certifications@rim.com (519) 888-7465 ext. 72921 | Research In Motion 295 Phillip Street Waterloo, Ontario Canada N2L 3W8 www.rim.com www.blackberry.com |

Contents

| | |
|---|----|
| Introduction | 1 |
| Cryptographic Module Specification..... | 2 |
| Cryptographic Module Ports and Interfaces..... | 5 |
| Roles, Services, and Authentication | 6 |
| Physical Security..... | 8 |
| Cryptographic Keys and Critical Security Parameters | 9 |
| Self-Tests | 10 |
| Mitigation of Other Attacks | 12 |
| Glossary | 13 |

List of Tables

| | |
|---|----|
| Table 1. Security Functions | 2 |
| Table 2. Implementation of FIPS 140-2 Interfaces..... | 5 |
| Table 3. Module Services | 6 |
| Table 4. Role Selection by Module Service..... | 7 |
| Table 5. Cryptographic Keys and CSPs..... | 9 |
| Table 6. Module Self-Tests..... | 10 |
| Table 7. Attack Types..... | 12 |

List of Figures

| | |
|---|---|
| Figure 1. BlackBerry Solution Architecture..... | 1 |
| Figure 2. Physical Boundary..... | 4 |

Introduction

BlackBerry® is the leading wireless solution that allows users to stay connected to a full suite of applications, including email, phone, enterprise applications, Internet, SMS, and organiser 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 architecture is shown in the following figure.

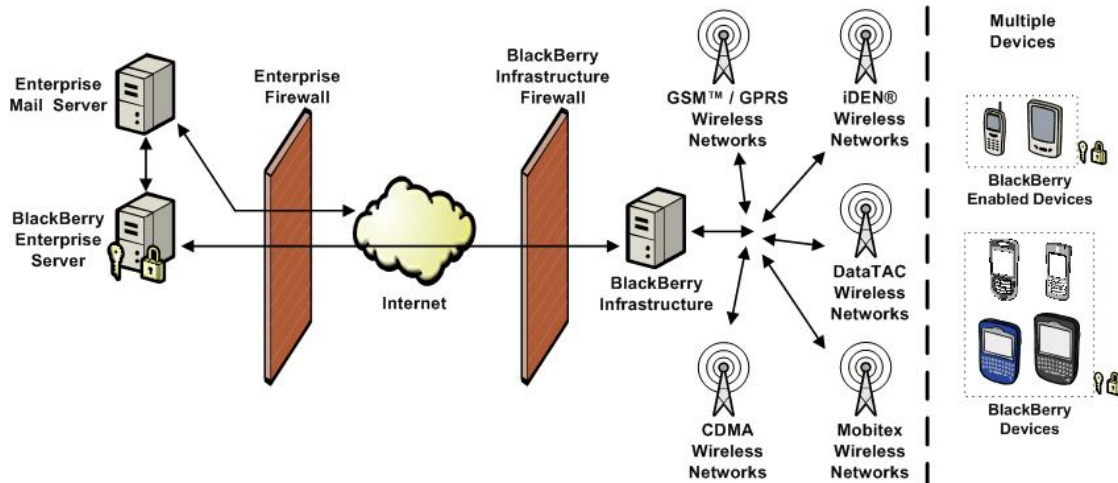


Figure 1. BlackBerry Solution Architecture

BlackBerry devices are built on industry-leading wireless technology, allowing users to receive email and information automatically with no need to request for delivery. Additionally, users are notified when new information arrives, making it easier to stay informed.

BlackBerry devices also provide an intuitive user experience. Users simply click on an email address, telephone number, or URL inside a message to automatically begin composing the new email, make the call, or link to the web page. BlackBerry device users can also easily navigate through icons, menus, and options with the roll-and-click trackwheel or trackball, and quickly compose messages or enter data using the device keyboard.

Each BlackBerry device¹ contains the BlackBerry Cryptographic Kernel, a firmware module that provides the cryptographic functionality required for basic operation of the device. The BlackBerry Cryptographic Module meets the requirements of the FIPS 140-2 Security Level 1.

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

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

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

¹ Excludes RIM 850™, RIM 950™, RIM 857™, and RIM 957™ wireless handheld devices.

Cryptographic Module Specification

Security Functions

The cryptographic module is a firmware module that implements the following FIPS-Approved security functions²:

Table 1. Security Functions

| Algorithm | Description | Version | Certificate Number |
|--|--|----------|--------------------|
| AES-256 ASM Code | Encrypt and Decrypt, as specified in FIPS 197. The implementation supports the CBC and CTR modes of operation. | 3.8.5.51 | #947 |
| AES-256 Native Code | Encrypt and Decrypt, as specified in FIPS 197. The implementation supports the CBC and CTR modes of operation. | 3.8.5.51 | #946 |
| Triple DES | Encrypt and Decrypt, as specified in FIPS 46-3. The implementation supports the CBC mode of operation. | 3.8.5.51 | #750 |
| SHA-1, SHA-256 & SHA-512 | as specified in FIPS 180-3. | 3.8.5.51 | #921 |
| HMAC-SHA-1, HMAC-SHA-256 & HMAC-SHA-512 | as specified in FIPS 198. | 3.8.5.51 | #526 |
| FIPS 186-2 RNG | as specified in FIPS 186-2. The implementation uses SHA-1 as the function G | 3.8.5.51 | #536 |

² A security function is FIPS-Approved if it is explicitly listed in *FIPS 140-2 Annex A: Approved Security Functions for FIPS PUB 140-2*.

| | | | |
|-------------------|--|----------|------|
| RSA PKCS#1 | Signature verification, as specified in PKCS #1, version 2.1 | 3.8.5.51 | #456 |
| ECDSA | Signature verification, as specified in FIPS 186-2 and ANSI X9.62. The implementation supports elliptic curve K-571. | 3.8.5.51 | #118 |

The module implements the following non-Approved security functions that, per *FIPS 140-2 Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2*, may presently be used in a FIPS-Approved mode of operation:

- **EC Diffie-Hellman** (key agreement, key establishment methodology provides 256 bits of encryption strength), as specified in IEEE P1363 Draft 13. The implementation supports elliptic curves P-521 and K-571.
- **ECMQV** (key agreement, key establishment methodology provides 256 bits of encryption strength), as specified in IEEE P1363 Draft 13. The implementation supports elliptic curves P-521 and K-571.

Modes of Operation

The module does not have a non-Approved mode of operation and, consequently, always operates in a FIPS-Approved mode of operation.

Conformance Testing and FIPS-Compliance

For the purposes of FIPS 140-2 conformance testing, the module was executed on the BlackBerry Storm 9500, per FIPS 140-2 Implementation Guidance G.5, remains FIPS-compliant when executed on other BlackBerry devices.

Conformance testing was performed using BlackBerry OS version 4.7. In order for the module to remain validated on a specific handheld device, both the module and the tested operating platform shall be ported to any device unchanged.

Cryptographic Boundary

The physical boundary of the module is the physical boundary of the BlackBerry device that executes the module and is shown in the following figure. Consequently, the embodiment of the module is multiple-chip standalone.

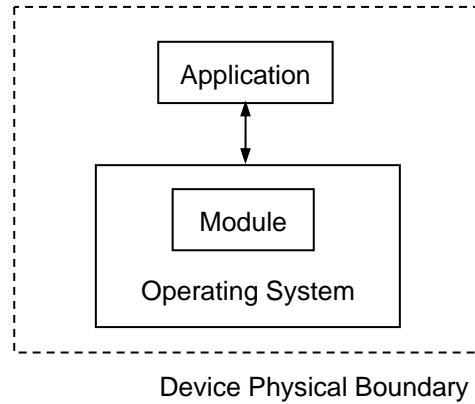


Figure 2. Physical Boundary

Determining the Module Version

The operator may determine the version of the module on a BlackBerry device by performing the following operations:

1. Navigate to the **Options** list.
2. Click the **About** item.
3. The About screen appears and displays the module version, e.g. "Cryptographic Kernel v3.8.5.51".

Cryptographic Module Ports and Interfaces

The module ports correspond to the physical ports of the BlackBerry device executing the module, and the module interfaces correspond to the logical interfaces to the module. 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 | Keyboard, microphone, USB port, headset jack, wireless modem, Bluetooth® wireless radio | Input parameters of module function calls |
| Data Output | Speaker, USB port, headset jack, wireless modem, Bluetooth wireless radio | Output parameters of module function calls |
| Control Input | Keyboard, USB port, trackwheel, trackball, touchscreen, menu button, escape button, backlight button, phone button, end button | Module function calls |
| Status Output | USB port, Primary LCD screen, LED | Return codes of module function calls |
| Power Input | USB port | Not supported |
| Maintenance | Not supported | Not supported |

Roles, Services, and Authentication

Roles

The module supports a User and Crypto Officer role. 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 of operation.

Services

The services described in the following table are available to the operator.

Table 3. Module Services

| Service | Description |
|-----------------------|--|
| Reset | Resets the module. The module may be reset by pressing the Alt + Right Shift + Backspace key combination or power cycling the module. |
| View Status | Displays the status of the module. |
| Perform Key Agreement | Establishes a secure channel to the module utilizing ECDH and ECMQV key agreement algorithms in transport of the new Master Key that is created outside the cryptographic boundary. |
| Inject PIN Master Key | Replaces the existing PIN Master Key with a new PIN Master Key. The new PIN Master Key is created outside the cryptographic boundary and is encrypted for input into the module for this service. |
| Generate Session Key | Generates a Session Key or a PIN Session Key. This service is performed automatically on behalf of the operator during the Encrypt Data service. |
| Encrypt Data | Encrypts data that is to be sent from the device. A Session Key is automatically generated via the Generate Session Key service and used to encrypt the data. The Session Key is encrypted with the Master Key and then the encrypted data and encrypted Session Key are ready for transmission. |
| Decrypt Data | Decrypts data that has been received by the device. The encrypted Session Key is decrypted with the Master Key and is then used to decrypt the data. This service is performed automatically on behalf of the operator. |
| Generate HMAC | Generates a message authentication code. |
| Perform Self-Tests | Executes the module self-tests. |
| Verify Signature | Verifies the digital signature of an IT policy received by the device. This service is performed automatically on behalf of the operator. |
| Wipe Handheld | Zeroizes device keys and user data present on device. |

Authentication

The module does not support operator authentication. Roles are implicitly selected based on the service performed by the operator. Implicit role selection is summarised in the following table, as are the keys and critical security parameters (CSPs) that are affected by each service.

Table 4. Role Selection by Module Service

| Service | Role Implicitly Selected | Affected Keys and CSPs | Access to Keys and CSPs |
|-----------------------|--------------------------|----------------------------------|-------------------------|
| Reset | User | N/A | N/A |
| View Status | User | N/A | N/A |
| Perform Key Agreement | Crypto Officer | ECC Key Pair | Execute |
| | | Master Key | Write |
| Inject PIN Master Key | Crypto Officer | PIN Master Key | Write |
| Generate Session Key | User | Session Key / PIN Session Key | Write |
| Encrypt Data | User | Master Key / PIN Master Key | Execute |
| | | Session Key / PIN Session Key | Execute |
| Decrypt Data | User | Master Key / PIN Master Key | Execute |
| | | Session Key / PIN Session Key | Execute |
| Generate HMAC | User | HMAC Key | Execute |
| Perform Self-Tests | User | Firmware Integrity Key | Execute |
| Verify Signature | User | ECC Public Key | Execute |
| Wipe Handheld | Crypto Officer | All | Write |

Physical Security

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

Cryptographic Keys and Critical Security Parameters

The following table describes the cryptographic keys, key components, and CSPs utilised by the module.

Table 5. Cryptographic Keys and CSPs

| Key / CSP | Description |
|-----------------|--|
| Master Key | <p>A Triple DES or AES-256 key used to encrypt and decrypt Session Keys. The Master Key is always generated outside the cryptographic boundary. The Key may be input into the module:</p> <ul style="list-style-type: none"> in plaintext as parameters to an API call when connected directly to the USB port of a workstation operating BlackBerry Desktop Manager, or encrypted by the current Master Key if utilizing key agreement with the BlackBerry Enterprise Server. |
| Session Key | A Triple DES or AES-256 key used to encrypt and decrypt data. The module generates Session Keys using the implemented FIPS 186-2 RNG. |
| PIN Master Key | <p>A Master Key that is specifically a Triple DES key used to encrypt and decrypt PIN Session Keys. The PIN Master Key is generated outside the cryptographic boundary. The Key may be input into the module:</p> <ul style="list-style-type: none"> in plaintext as parameters to an API call when connected directly to the USB port of a workstation operating BlackBerry Desktop Manager, or encrypted by the current Master Key if utilizing key agreement with the BlackBerry Enterprise Server. |
| PIN Session Key | A Session Key that is specifically a Triple DES key used to encrypt and decrypt data for PIN messaging. The module generates PIN Session Keys using the implemented FIPS 186-2 RNG. |
| ECC Key Pair | A key pair used to perform key agreement during Master Key transport over elliptic curves. |
| ECC Session Key | An ECC session key, that is specifically a short lived ephemeral key, is used during key agreement during Master Key transport and is zeroized after use. |
| ECC Public Key | A public key used to verify digital signatures over elliptic curves |
| HMAC Key | A key used to calculate a message authentication code using the HMAC algorithm. |

Key Zeroization

The BlackBerry security solution provides multiple protective features to ensure algorithmic keys and key components are protected. Similarly, data, and specifically key, removal through zeroization is built in as an integral part of the BlackBerry security solution. A user may also request a zeroization at anytime by navigating to the **Options** list and selecting “Wipe Handheld” via the **Security Options** → **General Settings**. The BlackBerry Enterprise Server administrator may also zeroize the device remotely to wipe all device data and keys.

Furthermore, session keys that are created per datagram are destroyed after each data fragment is sent.

Self-Tests

The module implements the self-tests described in the following table.

Table 6. Module Self-Tests

| Test | Description |
|-------------------------|---|
| Firmware Integrity Test | The module implements an integrity test for the module by verifying its 1024-bit RSA signature. The firmware integrity test passes if and only if the signature verifies successfully using the Firmware Integrity Key. |
| AES-256 CAT | <p>The module implements a compare answer test (CAT) for the AES-256 variants. Each AES implementation takes the same test data and same test key to perform an encryption operation. The result of each encryption operation is then compared to each other to verify that they were able to calculate the same result. If the results are the same, the test passes. If the results are different, the encrypt test fails.</p> <p>The module then performs a compared test for decryption using known encrypted test data and test key where each implementation is given the same key and data and performs a decryption operation. The results of each decryption operation from the C++ and assembler implementations are then compared against the calculated results. If both implementations are able to calculate the same result, the test passes. If they do not calculate the same result, then the test fails.</p> |
| Triple DES CBC KAT | The module implements a KAT for Triple DES in the CBC mode of operation. The test passes if and only if the calculated output equals the expected output. |
| SHA-1 KAT | The module implements a KAT for SHA-1. The KAT passes if and only if the calculated output equals the expected output. |
| SHA-256 KAT | The module implements a KAT for SHA-256. The KAT passes if and only if the calculated output equals the expected output. |
| SHA-512 KAT | The module implements a KAT for SHA-512. The KAT passes if and only if the calculated output equals the expected output. |
| HMAC SHA-1 KAT | The module implements a KAT for HMAC SHA-1. The KAT passes if and only if the calculated output equals the expected output. |
| HMAC SHA-256 KAT | The module implements a KAT for HMAC SHA-256. The KAT passes if and only if the calculated output equals the expected output. |
| HMAC SHA-512 KAT | The module implements a KAT for HMAC SHA-512. The KAT passes if and only if the calculated output equals the expected output. |
| RSA Verify KAT | The module implements a KAT for RSA signature verification. The test passes if and only if the calculated output equals the expected output. |
| ECDSA Verify KAT | The module implements a KAT for ECDSA signature verification. The test passes if and only if the calculated output equals the expected output. |
| FIPS 186-2 RNG KAT | The module implements a KAT for the FIPS 186-2 RNG. The KAT passes if and only if the calculated output equals the expected output. |
| Continuous RNG Test | The module implements a continuous RNG test, as specified in FIPS 140-2, for the implemented FIPS 186-2 RNG. |

All self-tests, except the Continuous RNG Test, are executed during power-up without requiring operator input or action. The Firmware Integrity Test is the first self-test executed during power-up.

Invoking the Self-Tests

The operator may invoke the power-up self-tests by resetting the module via the Reset service.

The operator may also invoke all of the self-tests with the exception of the Firmware Integrity Test and Continuous RNG test by performing the following operations:

1. Navigate to the Security options screen.
2. Click the **General Settings** option item.
3. Depending on the handheld model, click the trackwheel or trackball to open the General Settings options menu.
4. In the menu, click **Verify Security Software**.

When the self-tests are executed in this manner, the module displays the list of self-tests that are being executed and a pass/fail status upon completion.

Mitigation of Other Attacks

The module is designed to mitigate multiple side-channel attacks specific to the AES algorithm. Mitigation of these attacks is accomplished through the execution of table masking, splitting, and stirring manoeuvres designed to aid in the protection of cryptographic keys and plain text data at all points during the encryption, decryption, and self-test operations.

The following table describes the types of attacks the module mitigates.

Table 7. Attack Types

| Attack type | Description |
|--------------|--|
| Side-Channel | <ul style="list-style-type: none">attempts to exploit physical properties of the algorithm implementation using Power Analysis (for example, SPA and DPA) and Electro-Magnetic Analysis (for example, SEMA and DEMA)attempts to determine the encryption keys that a device uses by measuring and analyzing the power consumption, or electro-magnetic radiation, emitted by the device during cryptographic operations |

Glossary

| | |
|-------|---|
| AES | Advanced Encryption Standard |
| ANSI | American National Standards Institute |
| CBC | Cipher block chaining |
| CSP | Critical security parameter |
| DES | Data Encryption Standard |
| EC | Elliptic curve |
| ECC | Elliptic curve cryptography |
| ECDSA | Elliptic curve Digital Signature Algorithm |
| ECMQV | Elliptic curve Menezes, Qu, Vanstone |
| FIPS | Federal Information Processing Standard |
| HMAC | Keyed-hashed message authentication code |
| IEEE | Institute of Electrical and Electronics Engineers |
| KAT | Known answer test |
| LCD | Liquid crystal display |
| LED | Light emitting diode |
| PIN | Personal identification number |
| PKCS | Public Key Cryptography Standard |
| PUB | Publication |
| RIM | Research In Motion |
| RNG | Random number generator |
| RSA | Rivest, Shamir, Adleman |
| SHA | Secure Hash Algorithm |
| SHS | Secure Hash Standard |
| SMS | Short Messaging Service |
| URL | Uniform resource locator |
| USB | Universal serial bus |