FIPS 140-2 Security Policy

BlackBerry Cryptographic Tool Kit, Versions 6.0 and 6.0.2

Document version 1.0

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.

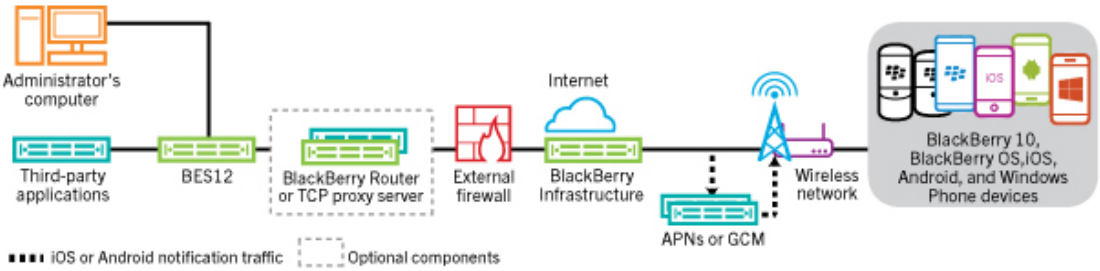


Figure 1. BlackBerry Enterprise Service 12 architecture

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

The BlackBerry Cryptographic Tool Kit, 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**
- **Digital signature verification**
- **Elliptic curve key agreement**

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

The BlackBerry Cryptographic Tool Kit 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**

1 Cryptographic module specification

The BlackBerry Cryptographic Tool Kit 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

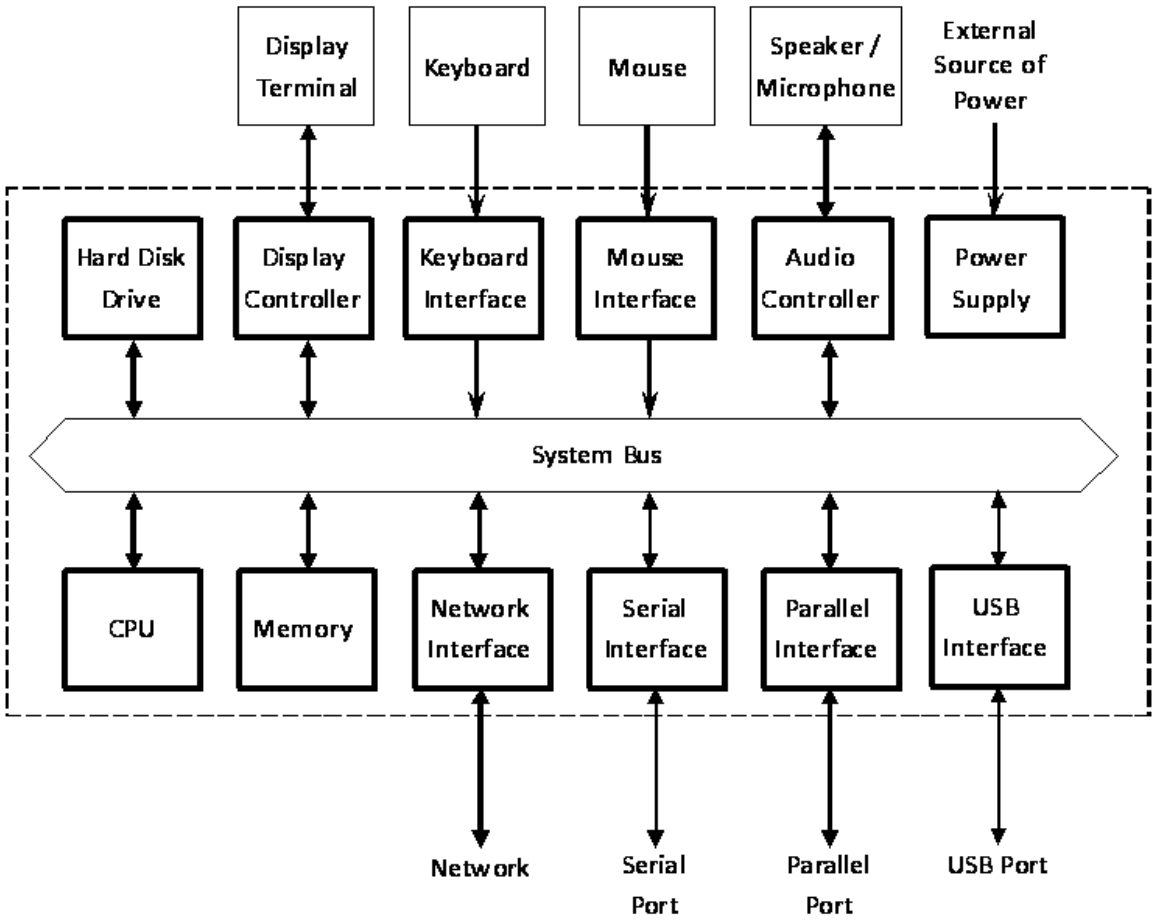
The general, computer hardware component consists of the following devices:

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

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.



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:

For version 6.0:

1. **QNX® Neutrino® 6.6, ARMv7 (Binary compatible to QNX Neutrino 6.5)**
2. **QNX Neutrino 6.5 x86**
3. **Red Hat® Linux® AS 5.6 32-bit x86 (Binary compatible to AS 4.x/5.0-5.5)**
4. **Red Hat Linux AS 5.6 64-bit x86 (Binary compatible to AS 4.x/5.0-5.5)**

For version 6.0.2:

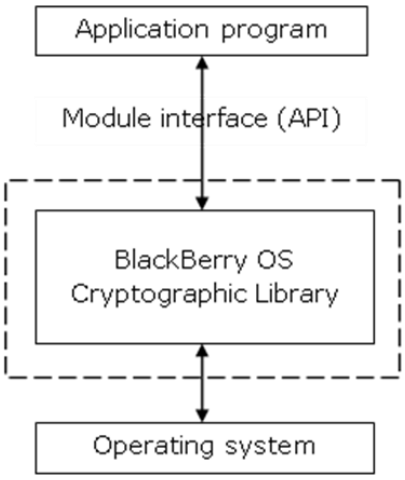
1. **Android 4.4.2, ARMv7**
2. **Android 4.0.4, x86**
3. **iOS version 6.1.4, ARMv7**
4. **Windows Phone® 8.0, ARMv7**
5. **Windows® 7 Enterprise, 64-bit x86**

The BlackBerry Cryptographic Tool Kit is also suitable for any manufacturer's platform that has compatible processors, equivalent or larger system configurations, and compatible OS versions. For example, an identical BlackBerry Cryptographic Tool Kit can be used on any compatible Linux or Windows for x86 processors, or iOS for ARMv7 processors. The BlackBerry Cryptographic Tool Kit 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 Cryptographic Tool Kit provides services to the C computer language users in an object format. A single source code base is used for all identified computer hardware and operating systems.

The interface into the BlackBerry Cryptographic Tool Kit 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.



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 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	Module interfaces
Data Input	**Keyboard, touch screen, microphone, USB port, headset jack, wireless modem, and Bluetooth® wireless radio**	**Input parameters of module function calls**
Data Output	**Speaker, USB port, headset jack, wireless modem, and Bluetooth wireless radio**	**Output parameters of module function calls**
Control Input	**Keyboard, touch screen, USB port, trackball, BlackBerry button, escape button, backlight button, and phone button**	**Module function calls**
Status Output	**USB port, primary LCD screen, and LED**	**Return codes of module function calls**
Power Input	**USB port**	**Initialization function**
Maintenance	**Not supported**	**Not supported**

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**	**X**
Deinitialization	**X**	**X**
Self-tests	**X**	**X**
Show status	**X**	**X**
Symmetric ciphers (AES, TDES)		
Key generation	**X**	**X**
Encrypt	**X**	**X**
Decrypt	**X**	**X**
Key zeroization	**X**	**X**
Hash algorithms and message authentication (SHA, HMAC)		
Hashing	**X**	**X**
Message authentication	**X**	**X**
Random number generation (pRNG)		
Instantiation	**X**	**X**
Seeding	**X**	**X**
Request	**X**	**X**
CSP/Key zeroization	**X**	**X**

Services	Crypto Officer	User
Digital signature (DSA, ECDSA, RSA)		
Key pair generation	**X**	**X**
Sign	**X**	**X**
Verify	**X**	**X**
Key zeroization	**X**	**X**
Key establishment (DH, ECDH, ECMQV, RSA)		
Key pair generation	**X**	**X**
Shared secret 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 function

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

Table 4. Supported cryptographic algorithms

	Algorithm	FIPS Approved or Allowed	Certificate number
Block ciphers	**TDES (ECB, CBC, CFB64, OFB64) [FIPS 46-3]**	**X**	**#1159, #1773**
	AES (ECB, CBC, CFB128, OFB128, CTR, CCM, GCM, CMAC, XTS) [FIPS 197]	**X**	**#1789, #3029**

	Algorithm	FIPS Approved or Allowed	Certificate number
	AES EAX [ANSI C12.22]		
	DES (ECB, CBC, CFB64, OFB64)		
	DESX (ECB, CBC, CFB64, OFB64)		
	AES (CCM*) [ZigBee 1.0.x]		
	ARC2 (ECB, CBC, CFB64, OFB64) [RFC 2268]		
Stream cipher	**ARC4**		
Hash functions	**SHA-1 [FIPS 180-3]**	**X**	**#1571, #2530**
	SHA-224 [FIPS 180-3]	**X**	**#1571, #2530**
	SHA-256 [FIPS 180-3]	**X**	**#1571, #2530**
	SHA-384 [FIPS 180-3]	**X**	**#1571, #2530**
	SHA-512 [FIPS 180-3]	**X**	**#1571, #2530**
	MD5 [RFC 1321]		
	MD4 [RFC 1320]		
	MD2 [RFC 1115]		
	AES MMO [ZigBee 1.0.x]		
Message authentication	**HMAC-SHA-1 [FIPS 198]**	**X**	**#1054, #1914**
	HMAC-SHA-224 [FIPS 198]	**X**	**#1054, #1914**
	HMAC-SHA-256 [FIPS 198]	**X**	**#1054, #1914**
	HMAC-SHA-384 [FIPS 198]	**X**	**#1054, #1914**

	Algorithm	FIPS Approved or Allowed	Certificate number
	HMAC-SHA-512 [FIPS 198]	**X**	**#1054, #1914**
	HMAC-MD5 [RFC 2104]		
	AES-XCBC-MAC [RFC 3566]		
pRNG	**DBRG [NIST SP 800-90]**	**X**	**#127, #579**
	ANSI X9.62 RNG [ANSI X9.62]	**X**	**#949, #1310**
	ANSI X9.31 RNG [ANSI X9.31]	**X**	**#949, #1310**
Digital signature	**DSS [FIPS 186-3]**	**X**	**#563, #891**
	ECDSA [FIPS 186-3, ANSI X9.62]	**X**	**#242, #553**
	RSA PKCS1 v1.5 [FIPS 186-3, PKCS#1 v2.1]	**X**	**#894, #1574**
	RSA PSS [FIPS 186-3, PKCS#1 v2.1]	**X**	**#894, #1574**
	ECNR [IEEE 1363]		
	ECQV		
Key agreement	**DH [NIST SP 800-56A] Security strength >= 112 bits**	**X**	**#25, #50**
	DH [NIST SP 800-56A] Security strength < 112 bits		
	ECDH [NIST SP 800-56A] Component (ECC CDH)	**X**	**#25, #50 #7, #367**
	ECMQV [NIST SP 800-56A]	**X**	**#25, #50**
	ECPVS [ANSI X9.92]		

	Algorithm	FIPS Approved or Allowed	Certificate number
	ECSPEKE [IEEE 1363.2]		
Key wrapping	**RSA PKCS1 v1.5 [PKCS#1 v2.1]**	**X**	
	RSA OAEP [NIST SP 800-56B]		
	RSA KEM [ANSI X9.44]		
	ECIES [ANSI X9.63]		

The DES, DESX, AES CCM* (CCM Star) and EAX modes, ARC2, ARC4, MD5, MD4, MD2, AES MMO, HMAC-MD5, AES-XCBC-MAC, ECNR, ECQV, ECIES, ECPVS, ECSPEKE, key establishment (key wrapping) techniques, RSA OAEP and RSA KEM, and DH with strength < 112 bits are supported as non FIPS Approved algorithms. To operate the module in compliance with FIPS, these algorithms should not be used.

Note: Until December 31, 2015, the use of 2-Key Triple-DES for encryption is restricted, after which time it will become disallowed for encryption. When used for encryption, the total number of blocks of data encrypted with the same cryptography key shall not be greater than 220. The use of 3-Key Triple DES is strongly encouraged. For more information, see NIST SP 800-131A.

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**	**Create, Read, Use**
TDES	**Key**	**168 bits**	**112 bits**	**Create, Read, Use**
HMAC	**Key**	**160 to 512 bits**	**128 to 256 bits**	**Use**
pRNG (ANSI X9.62, ANSI X9.31, DRBG)	**Seed key, seed**	**160 bits**	**80 bits**	**Use**

Algorithm	Key and CSP	Key size	Security strength	Access
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**	**2048 to 15360 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; non-compliant less than 112-bits of encryption strength).**
- **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).**
- **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.**

3.3 Operator authentication

The BlackBerry Cryptographic Tool Kit 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:

- **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.**

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.

6 Operational environment

The BlackBerry Cryptographic Tool Kit 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, Windows, Android, iOS, or QNX provide such operational environments.

7 Cryptographic key management

The BlackBerry Cryptographic Tool Kit 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 Tool Kit provides FIPS 140-2 compliant key generation. The underlying random number generation uses a FIPS Approved method, DRBG (Hash, HMAC, Cipher), ANSI X9.62 RNG (SHA-1), or ANSI X9.31 RNG (AES-128, 192, and 256).

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 Tool Kit 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 163 bits to 521 bits that provides 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 PKCS1 v1.5: The RSA PKCS v1.5 key wrapping implementation supports modulus sizes from 512 bits to 15360 bits that provides between 56 bits 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

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 Tool Kit is a low-level cryptographic toolkit; therefore, it does not provide key storage.

7.5 Key zeroization

The BlackBerry Cryptographic Tool Kit provides zeroizable interfaces that implement zeroization functions; for more information, see Table 3. Zeroization of keys and SPs must be performed by calling the destroy functions of the objects when they are no longer needed; otherwise, the BlackBerry Cryptographic Tool Kit will not function.

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)	**KATs are performed on TDES, AES, ASE GCM, SHS (using HMAC-SHS), HMAC-SHS, DRBG, ANSI X9.62 RNG, ANSI X9.31 RNG, RSA Signature Algorithm, and KDF. For DSA and ECDSA, a Pair-wise Consistency Test is used.** **For DH, ECDH, ECMQV, the underlying arithmetic implementations are tested using DSA and ECDSA tests.**
Software integrity test	**The software integrity test deploys ECDSA signature validation 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 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 reception of a DH, ECDH, or ECMQV key generation, the SP 800-56A conformant computation is performed.

8.4 Failure of self-tests

Self-test failure places the cryptographic module in the Error state, wherein no cryptographic operations can be performed. If any self-test fails, the cryptographic module will output error code and enter the Error state.

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

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

10 Mitigation of other attacks

The BlackBerry Cryptographic Tool Kit 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.

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**

Acronym	Full term
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
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

Acronym	Full term
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, 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

Appendix C Crypto Office and User Guide

C.1 Installation

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

Note: Place the object in an appropriate location on the computer hardware for your development environment.

C.1.2 Uninstalling the cryptographic module

Remove the object from the computer hardware.

C.2 Commands

C.2.1 Initialization

The `sbg_FIPS140Initialize()` function runs a series of 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, a value of `SB_SUCCESS` is returned and the module is enabled.

C.2.2 Deinitialization

The `sbg_FIPS140Deinitialize()` function deinitializes the module.

C.2.3 Self-tests

The `sbg_FIPS140RunTest()` function runs a series of self-test and returns `SB_SUCCESS` 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 module is disabled. Section C 3 in this appendix describes how to recover from the disabled state.

C.2.4 Show Status

The `sbg_FIPS140GetState()` function returns the current state of the module.

C.3 When the cryptographic module is disabled

When BlackBerry Cryptographic Tool Kit becomes disabled, attempt to bring the module back to the Installed/Uninitialized state by calling `sbg_FIPS140Deinitialize()` and then to initialize the module by calling `sbg_FIPS140Initialize()`. If the initialization is successful, the module is recovered. If this attempt fails, uninstall the module and reinstall it. If the module is initialized successfully after this reinstallation, the recovery is successful. A failed recovery attempt indicates a fatal error. Contact BlackBerry Support immediately.

Document and contact information

Version	Date	Author	Reason for revision
1.0	**February 18, 2015**	**Randy Eyamie**	**Original release**

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**