FIPS 140-2 Security Policy

BlackBerry OS Cryptographic Library Versions 5.6, 5.6.1 and 5.6.2

Document Version 1.6

BlackBerry Security Certifications, Research In Motion

www.blackberry.com

Document and contact information

Version	Date	Description
1.0	April 7, 2011	Document creation
1.1	June 14, 2011	Addressed CMVP Comments
1.2	July 20, 2011	Addressed CMVP Comments
1.3	May 22, 2012	Added software version 5.6.1
1.4	June 5, 2012	Corrected document version information
1.5	July 10, 2012	Added software version 5.6.2
1.6	February 6, 2013	Corrected OS reference in section 1.2 Computer Hardware and OS, to Reflect BlackBerry Tablet OS version 2.0

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

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, SMS, and organizer information. The BlackBerry solution is an integrated package that includes innovative software, advanced BlackBerry wireless devices and wireless network service, providing a seamless solution. The BlackBerry® Enterprise Solution architecture is shown in the following figure.

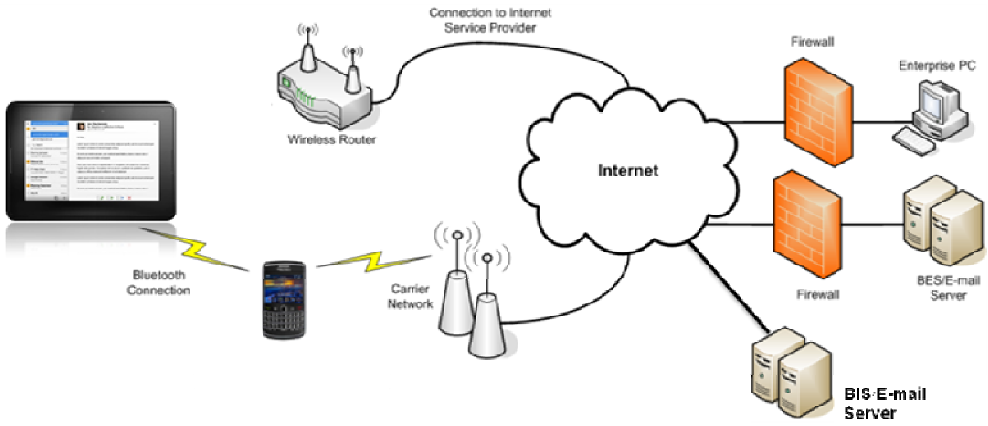

Figure 1. BlackBerry tablet connections

BlackBerry® PlayBook™ tablets are built on industry-leading wireless technology and use a powerful BlackBerry® Tablet OS. BlackBerry PlayBook tablets provide intuitive multi-tasking, allowing users to easily navigate the touch screen to switch between open applications, enjoy a PC-like web browsing experience with Adobe® Flash®, read rich media content and view HD video.. BlackBerry tablet users can access enterprise features by using a secure Bluetooth connection to supported BlackBerry® smartphones to the BlackBerry PlayBook tablet for real time access to PIM functionality (email, calendar, address book, task list and BBM™), and use the existing BlackBerry Enterprise Server connection to remotely access files and applications from an enterprise PC.

Each BlackBerry PlayBook tablet contains the BlackBerry OS Cryptographic Library, a software module that provides the cryptographic functionality required for basic operation of the device.

The BlackBerry OS Cryptographic Library, hereafter referred to as the cryptographic module or the 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 PlayBook tablet solution is available from http://www.blackberry.com/playbook.

The BlackBerry OS Cryptographic Library meets the requirements of the 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 OS Cryptographic Library is a multiple-chip stand-alone software cryptographic module in the form of a shared object (*libsbgse56.so.0.0*) that operates with the following components:

- Commercially available general-purpose computer hardware
- Commercially available operating system (OS) that runs on the computer hardware

1.1 Physical specifications

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

1. ARMv7 CPU (microprocessor)
2. Memory
   (a) Working memory is located on the RAM and contains the following spaces:
      i. Input/output buffer
      ii. Plaintext/ciphertext buffer
      iii. Control buffer

   Key storage is not deployed in this module.

   (b) Program memory is also located on RAM
3. Hard disk (or disks), including flush memory
4. Display controller, including the touch screen controller
5. Keyboard interface
6. Mouse interface, including the trackball interface
7. Audio controller
8. Network interface
9. Serial port
10. Parallel port
11. USB interface
12. Power supply

The configuration of this component is illustrated in Figure 2.

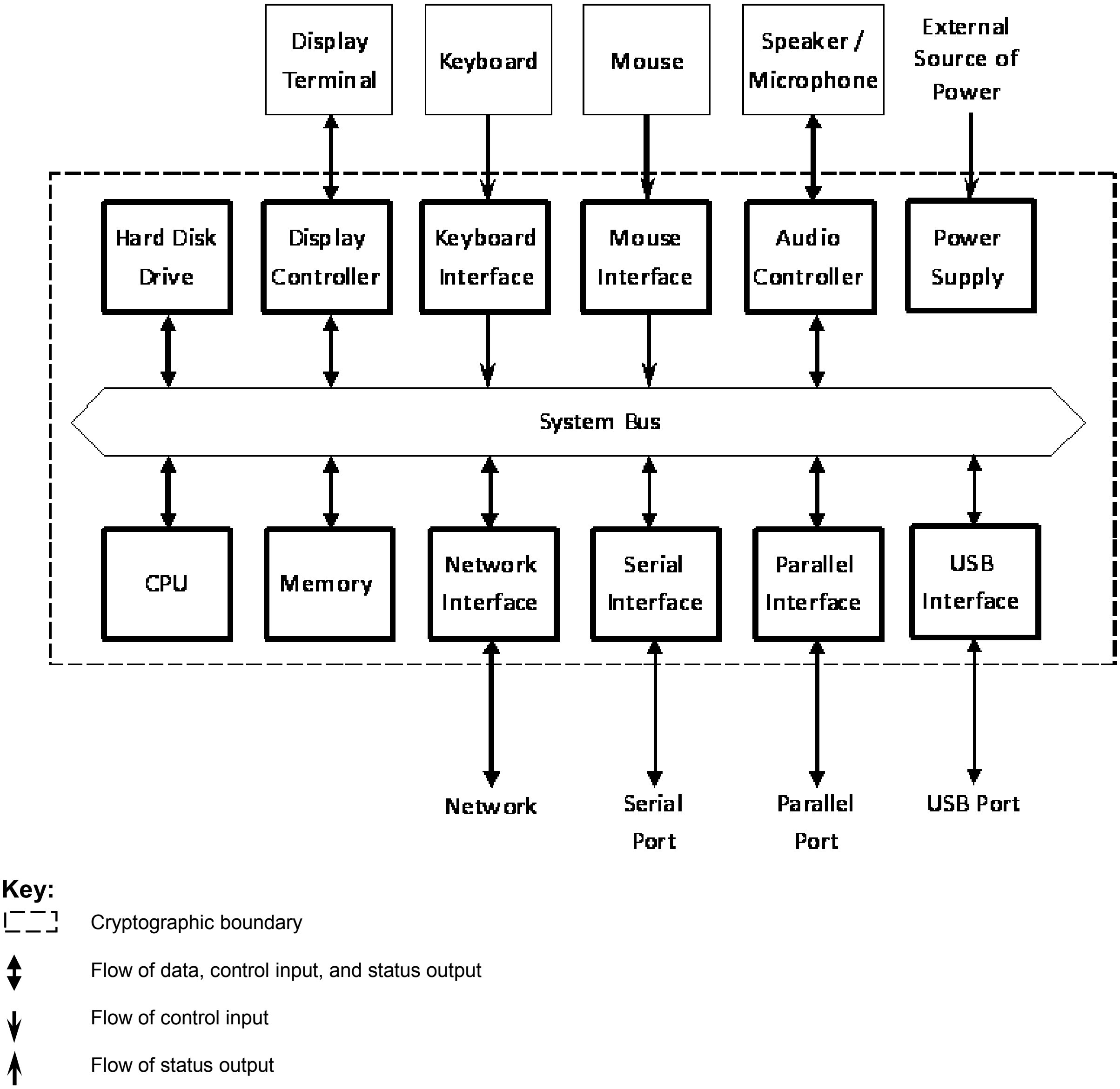

Figure 1: Cryptographic module hardware block diagram

1.2 Computer hardware and OS

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

BlackBerry Tablet OS version 2.0 (Binary compatible to BlackBerry Tablet OS version 1.0), ARMv7

The BlackBerry OS Cryptographic Library 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 OS Cryptographic Library can be used on any compatible BlackBerry Tablet OS for ARM processors. The BlackBerry OS Cryptographic Library will run on such platforms and OS versions while maintaining its compliance to the FIPS 140-2 Level 1 requirements.

1.3 Software specifications

The BlackBerry OS Cryptographic Library provides services to the C computer language users in a shared object format. A single source code base is used for all identified computer hardware and operating systems.

The interface into the BlackBerry OS Cryptographic Library is through Application Programmer's 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 (see Figure 3).

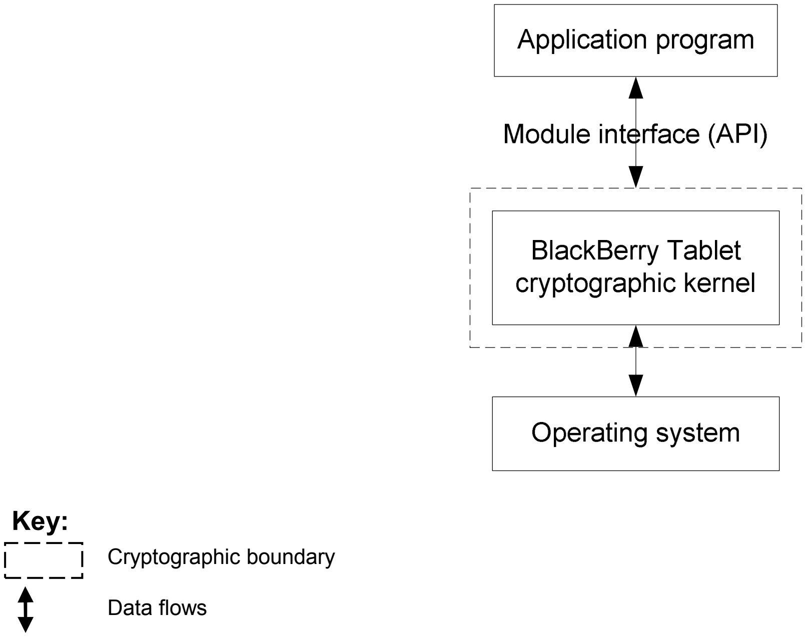

Figure 3: Cryptographic module software block diagram

2 Cryptographic Module Ports and Interfaces

The cryptographic module ports correspond to the physical ports of the BlackBerry tablet 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 a single-user mode.

Table 3. Module roles and services

Service	Crypto Officer	User
Initialization, etc.		
Initialization	X	X
Deinitialization	X	X
Self-tests	X	X
Show status	X	X
Symmetric Ciphers (AES and TDES)		
Key generation	X	X
Encrypt	X	X
Decrypt	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
Digital Signature (DSA, ECDSA, RSA)		
Key pair generation	X	X
Sign	X	X
Verify	X	X
Key Establishment (DH, ECDH, ECMQV, RSA)		

Service	Crypto Officer	User
Key pair generation	X	X
Shared secret generation	X	X
Wrap	X	X
Unwrap	X	X
Key Zeroization	X	X

In order to operate the module securely, it is the Crypto Officer's and the User's responsibility to confine calls to 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 OS Cryptographic Library supports many cryptographic algorithms. The set of cryptographic algorithms supported by the BlackBerry OS Cryptographic Library is shown in Table 4.

Table 4. Approved security functions

	Algorithm	FIPS Approved or Allowed	Certificate number
Block Ciphers	TDES (ECB, CBC, CFB64, OFB64 [FIPS 46-3]	X	#1053
	AES (ECB, CBC, CFB128, OFB128, CTR, CCM, GCM, CMAC, XTS) [FIPS 197]	X	#1608
	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	#1421
	SHA-224 [FIPS 180-3]	X	#1421

	Algorithm	FIPS Approved or Allowed	Certificate number
	SHA-256 [FIPS 180-3]	X	#1421
	SHA-384 [FIPS 180-3]	X	#1421
	SHA-512 [FIPS 180-3]	X	#1421
	MD5 [RFC 1321]		
	MD4 [RFC 1320]		
	MD2 [RFC 1115]		
Message Authentication	HMAC-SHA-1 [FIPS 198]	X	#944
	HMAC-SHA-224 [FIPS 198]	X	#944
	HMAC-SHA-256 [FIPS 198]	X	#944
	HMAC-SHA-384 [FIPS 198]	X	#944
	HMAC-SHA-512 [FIPS 198]	X	#944
	HMAC-MD5 [RFC 2104]		
pRNG	DRBG [NIST SP 800-90]	X	#81
	ANSI X9.62 RNG [ANSI X9.62]	X	#862
	ANSI X9.31 RNG [ANSI X9.31]	X	#862
Digital Signature	DSS [FIPS 186-3]	X	#499
	ECDSA [FIPS 186-3, ANSI X9.62]	X	#199
	RSA PKCS1 v1.5 [FIPS 186-3, PKCS #1 v2.1]	X	#790
	RSA PSS [FIPS 186-3, PKCS #1 v2.1]	X	#790
	ECNR [IEEE 1363]		
	ECQV		

	Algorithm	FIPS Approved or Allowed	Certificate number
Key Agreement	DH [NIST SP 800-56A]	X	#13
	ECDH [NIST SP 800-56A]	X	#13
	ECMQV [NIST SP 800-56A]	X	#13
Key Wrapping	RSA PKCS1 v1.5 [PKCS #1 v2.1]	X	
	RSA OAEP [NIST SP 800-56B]	X	
	ECIES [ANSI X9.63]		

The TDES, AES (ECB, CBC, CFB128, OFB128, CTR, CCM, GCM, CMAC, and XTS modes), SHS (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512), HMAC-SHS (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA256, HMAC-SHA-384, and HMAC-SHA-512), pRNG (ANSI X9.62, ANSI X9.31, and NIST SP 800-90), DSA, ECDSA, RSA PKCS #1 v1.5 Signature, RSA PSS algorithms, and NIST SP 800-56A Key Establishment techniques (key agreement), DH, ECDH, and ECMQV, have been validated to comply with FIPS.

The BlackBerry OS Cryptographic Library also supports a NIST SP 800-56B Key Establishment technique (key wrapping), RSA OAEP. In order to operate the module in compliance with FIPS, only these FIPS Approved or allowed algorithms should be used.

The DES, DESX, AES CCM* (CCM Star) mode, ARC2, ARC4, MD5, MD4, MD2, HMAC-MD5, ECNR, ECQV, ECIES, and RSA #1 v1.5 encryption algorithm are supported as non FIPS Approved algorithms. In order to operate the module in compliance with FIPS, these algorithms should not be used.

*Please be advised, that 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 cryptographic key shall not be greater than 2^{20}. **The use of 3-Key Triple-DES is strongly encouraged.** Please see NIST SP 800-131A for more information.*

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

Table 5. Role selection by module service

Algorithm	Key and SP	Key size	Strength	Access
AES	Key	128-256 bits	128-256 bits	Create, Read, Use
TDES	Key	168 bits	112bits	Create, Read, Use
HMAC	Key	160-512 bits	80-256 bits	Create, Read, Use

Algorithm	Key and SP	Key size	Strength	Access
pRNG (ANSI X9.62, ANSI X9.31, DRBG	Seed key, seed	160 bits	80 bits	Create, Read, Use
DSA	Key pair	1024-15360 bits	80-256 bits	Create, Read, Use
ECDSA	Key pair	163-521 bits	80-256 bits	Create, Read, Use
RSA signature	Key pair	1024-15360 bits	80-256 bits	Create, Read, Use
DH	static/ephemeral key pair	1024-15360 bits	80-256 bits	Create, Read, Use
ECDH	static/ephemeral key pair	163-521 bits	80-256 bits	Create, Read, Use
ECMQV	static/ephemeral key pair	163-521 bits	80-256 bits	Create, Read, Use
RSA key wrapping	key pair	1024-15360 bits	80-256 bits	Create, Read, Use

3.3 Operator authentication

The BlackBerry OS Cryptographic Library does not deploy an authentication mechanism. The operator implicitly selects the roles of Crypto Officer and User.

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 transition:

1. When the module is installed by the Crypto Officer, the module is in the Installed/Uninitialized state.
2. When the initialization command is applied to the module, that is, the module is loaded on the memory, turning to the Initialization state. Then, the module transits 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 idle; 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 self-tests have succeeded, the module can transit to the Crypto Officer/User state when an API function is called.
4. When the API function has completed successfully, the state transits back to Idle.
5. If the conditional test (continuous RNG test or pair-wise consistency test) fails, the state transits 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 deinitialization command is executed, the module returns to the Installed/Uninitialized state.

5 Physical Security

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

6 Operational Environment

The BlackBerry OS Cryptographic Library is to run in 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.

7 Cryptographic Key Management

The BlackBerry OS Cryptographic Library 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 (see Table 4).

7.1 Key generation

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

7.2 Key establishment

The BlackBerry OS Cryptographic Library provides the following FIPS Approved or allowed key establishment techniques [5]:

1. Diffie-Hellman (DH)
2. EC Diffie-Hellman (ECDH)
3. ECMQV
4. RSA PKCS1 v1.5
5. RSA OAEP

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. 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 1024 bits and above must be used to provide minimum of 80 bits of security in the FIPS mode. The RSA OAEP key wrapping implementation supports modulus sizes from 512 to 15360 bits that provides between 56 bits and 256 bits of security, where 1024 bits and above must be used to provide minimum of 80 bits of security in the FIPS mode. It is responsibility of the application to ensure that the appropriate key establishment techniques are applied to the appropriate keys.

7.3 Key entry and output

Keys must be imported to or exported from the cryptographic boundary in encrypted form using a FIPS Approved algorithm.

7.4 Key storage

The BlackBerry OS Cryptographic Library is a low-level cryptographic toolkit, so it does not provide key storage.

7.5 Zeroization of keys

The BlackBerry OS Cryptographic Library provides zeroizable interfaces that implement zeroization functions (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 OS Cryptographic Library will not function.

8 Self-Tests

8.1 Power-up tests

8.1.1 Tests upon power-up

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

1. **Known Answer Tests (KATs):**

   KATs are performed on TDES, AES, AES GCM, SHS (via HMAC-SHS), HMAC-SHS, DRBG,

   ANSI X9.62 RNG, ANSI X9.31 RNG, RSA Signature Algorithm, and KDF. For DSA and ECDSA, Pair-wise Consistency Test is used. For DH, ECDH, ECMQV, the underlying arithmetic implementations are tested using DSA and ECDSA tests.

2. **Software Integrity Test:**

   The software integrity test deploys ECDSA signature validation to verify the integrity of the module.

8.1.2 On-demand self-tests

The Crypto Officer or User may invoke on-demand self-tests by invoking a function, which is described in the Crypto Officer And User Guide in Appendix A.

8.2 Conditional tests

The continuous RNG test is executed on all RNG generated data, examining the first160 bits of each requested random generation for repetition. This examination ensures that the RNG is not stuck at any constant value. Also, upon each generation of a DSA, ECDSA, or RSA key pair, the generated key pair is tested of their correctness by generating a signature and verifying the signature on a given message as a Pair-wise Consistency Test. Upon reception of DH, ECDH, or ECMQV key pair, the full key validation is performed. SP 800-56A conformant computation is performed upon DH, ECDH, or ECMQV key generation.

8.3 Failure of self-tests

Failure of the self-tests 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 a document that was submitted to the testing laboratory. It 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

Crypto Officer Guide And User Guide is provided in Appendix A. This appendix outlines the operations for Crypto Officer and User to ensure the security of the module.

10 Mitigation of Other Attacks

The BlackBerry OS Cryptographic Library 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 leaks information concerning the secret modulus.

In order to mitigate this attack, the following is executed: The bases of exponentiation are randomized by a novel technique that requires no inversion to remove (unlike other blinding methods, for example, BSAFE Crypto-C User Manual v 4.2).

Note: remote timing attacks are practical:

http://crypto.stanford.edu/ dabo/papers/ssl-timing.pdf

10.2 Attack on biased private key of DSA

The standards for choosing ephemeral values in El-Gamal type signatures introduce a slight bias. Means to exploit these biases were presented to ANSI by D. Bleichenbacher.

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

Change Notice 1 of FIPS 186-2 is published to mitigate this attack:
http://csrc.nist.gov/CryptoToolkit/tkdigsigs.html

Appendix A Crypto Officer and User Guide

A.1 Installation

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

A.1.1 Installing

The Crypto Officer is responsible for the installation of the BlackBerry OS Cryptographic Library . Only the Crypto Officer is allowed to install the product.

Note: Place the shared object, *libsbgse56.so.0.0* in an appropriate location on the computer hardware for your development environment.

A.1.2 Uninstalling

Remove the shared object, *libsbgse56.so.0.0* from the computer hardware.

A.2 Commands

A.2.1 Initialization

sbg56 FIPS140Initialize()

This function 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, a value of *SB_SUCCESS* is returned and the module is enabled.

A.2.2 De-initialization

sbg56 FIPS140Deinitialize()

This function deinitializes the module.

A.2.3 Self-tests

sbg56 FIPS140RunTest()

This function runs a series of self-tests and returns *SB_SUCCESS* if the tests are successful. These tests examine the integrity of the shared object and the correct operation of the cryptographic algorithms. If these tests fail, the module is disabled. Section A.3 of this document describes how to recover from the disabled state.

A.2.4 Show status

sbg56 FIPS140GetState()

This function returns the current state of the module.

A.3 When module is disabled

When the BlackBerry OS Cryptographic Library becomes disabled, attempt to bring the module back to the Installed state by calling *sbg56 FIPS140Deinitialize(),* and then to initialize the module using *sbg56 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 by this reinstallation, the recovery is successful. If this recovery attempt fails, it indicates a fatal error. Please contact Research In Motion Support immediately.

Appendix B Acronyms

Introduction

This appendix lists the acronyms that are used in this document.

Acronyms

Acronym	Full term
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	application programming interface
CAT	compare answer test
CBC	cipher block chaining
CSP	critical security parameter
DEMA	differential electromagnetic analysis
DES	Data Encryption Standard
DPA	differential power analysis
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-hash message authentication code
IEEE	Institute of Electrical and Electronics Engineers
KAT	known answer test
LCD	liquid crystal display
LED	light-emitting diode
OS	operating system
PIN	personal identification number
PKCS	Public Key Cryptography Standard
PUB	Publication

Acronym	Full term
RIM	Research In Motion
RNG	random number generator
RSA	Rivest, Shamir and Adleman
SEMA	simple electromagnetic analysis
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMS	Short Message Service
SPA	simple power analysis
URL	Uniform Resource Locator
USB	Universal Serial Bus

Appendix C 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,* January 4, 2011.
3. NIST *Security Requirements For Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2,* June 14, 2007.
4. NIST *Security Requirements For Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2, Draft,* November 22, 2010.
5. NIST *Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2, Draft,* January 4, 2011.
6. NIST *Derived Test Requirements for FIPS 140-2, Draft,* January 4, 2011.
7. NIST *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program,* December 23, 2010.
8. NIST *Frequently Asked Questions for the Cryptographic Module Validation Program,* December 4, 2007.