FIPS 140-2 Security Policy

BlackBerry Cryptographic Library for Secure Work Space Version 1.0

Document Version 1.2

BlackBerry Security Certifications, Research In Motion

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, Internet, Short Messaging Service (SMS), and organizer information. BlackBerry is a totally integrated package that includes innovative software, advanced BlackBerry® smartphones and wireless network service, providing a seamless solution. The BlackBerry architecture is shown in the following figure.

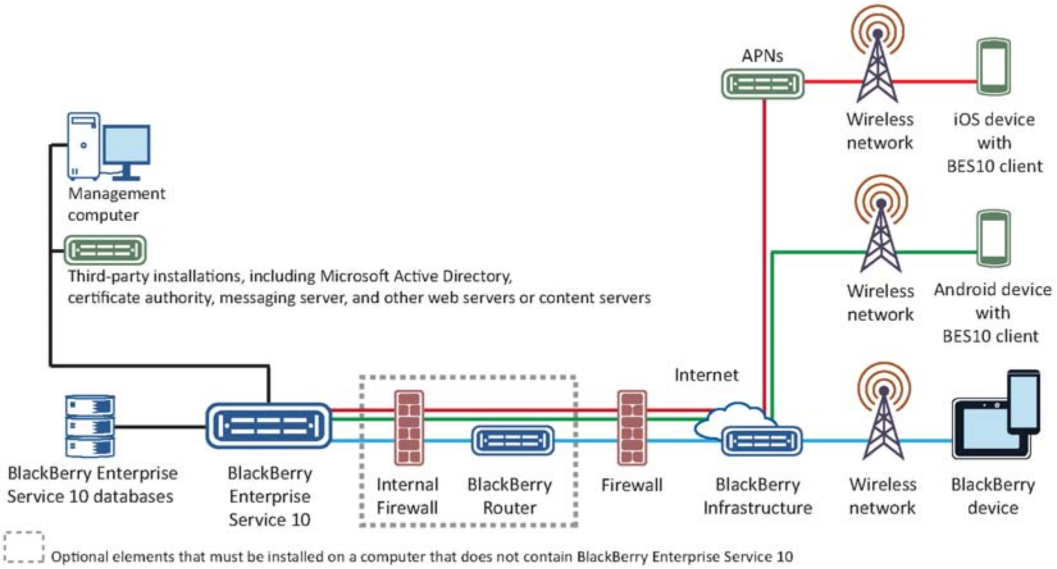

APNs
Wireless network
iOS device with BES10 client
Management computer
Third-party installations, including Microsoft Active Directory, certificate authority, messaging server, and other web servers or content servers
Wireless network
Android device with BES10 client
Internet
BlackBerry Enterprise Service 10 databases
BlackBerry Enterprise Service 10
Internal Firewall
BlackBerry Router
Firewall
BlackBerry Infrastructure
Wireless network
BlackBerry device
Optional elements that must be installed on a computer that does not contain BlackBerry Enterprise Service 10

Figure 1. BlackBerry Solution Architecture

BlackBerry Enterprise Service 10 helps you manage mobile devices for your organization. You can manage BlackBerry smartphones and BlackBerry PlayBook tablets, as well as iOS and Android devices, all from a unified interface. BlackBerry Enterprise Service 10 is designed to help protect business information, keep mobile workers connected with the information they need, and provide administrators with efficient tools that help keep business moving forward. BlackBerry Enterprise Service 10 includes the following components:

- The BlackBerry Device Service, providing advanced administration for BlackBerry 10 devices and BlackBerry PlayBook tablets
- The Universal Device Service, Providing advanced administration for iOS and Android devices
- The BlackBerry Management Studio, which provides a unified interface to administer common tasks for supported devices.

The Universal Device Service utilizes the Secure Work Space to control how certain work data is secured on iOS and Android devices. The work space feature uses the BlackBerry Infrastructure to create work and personal partitions on iOS and Android devices. You can control a user's work email, work apps, and work calendar settings from the Universal Device Service console. Configuring the work space allows you to keep all work data secure in your organization.

For more information on the BlackBerry solution, visit http://www.blackberry.com/.

The BlackBerry Cryptographic Library for Secure Work Space, hereafter referred to as cryptographic module or module, is a software module that provides the following cryptographic services to supported iOS and Android devices.

- symmetric/asymmetric cipher operation
- signature generation/verification
- hashing
- cryptographic key generation
- random number generation
- message authentication functions

The BlackBerry Cryptographic Library for Secure Work Space meets the overall requirements applicable to Level 1 security for FIPS 140-2 as shown in Table 1.

Table 1. Summary of Achieved Security Levels per FIPS 140-2 Section

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

1 Cryptographic Module Specification

The BlackBerry Cryptographic Library for Secure Work Space is a multiple-chip stand-alone software cryptographic module in the form of an object file (*fipscanister.o*) 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. Intel Xeon, AMD Opteron, or 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 flash memory
4. Display controller, including the touch screen controller
5. Keyboard interface
6. Mouse interface
7. Audio controller
8. Network interface
9. Serial port
10. USB interface
11. Power supply

The configuration of this component is illustrated in Figures 2 & 3.

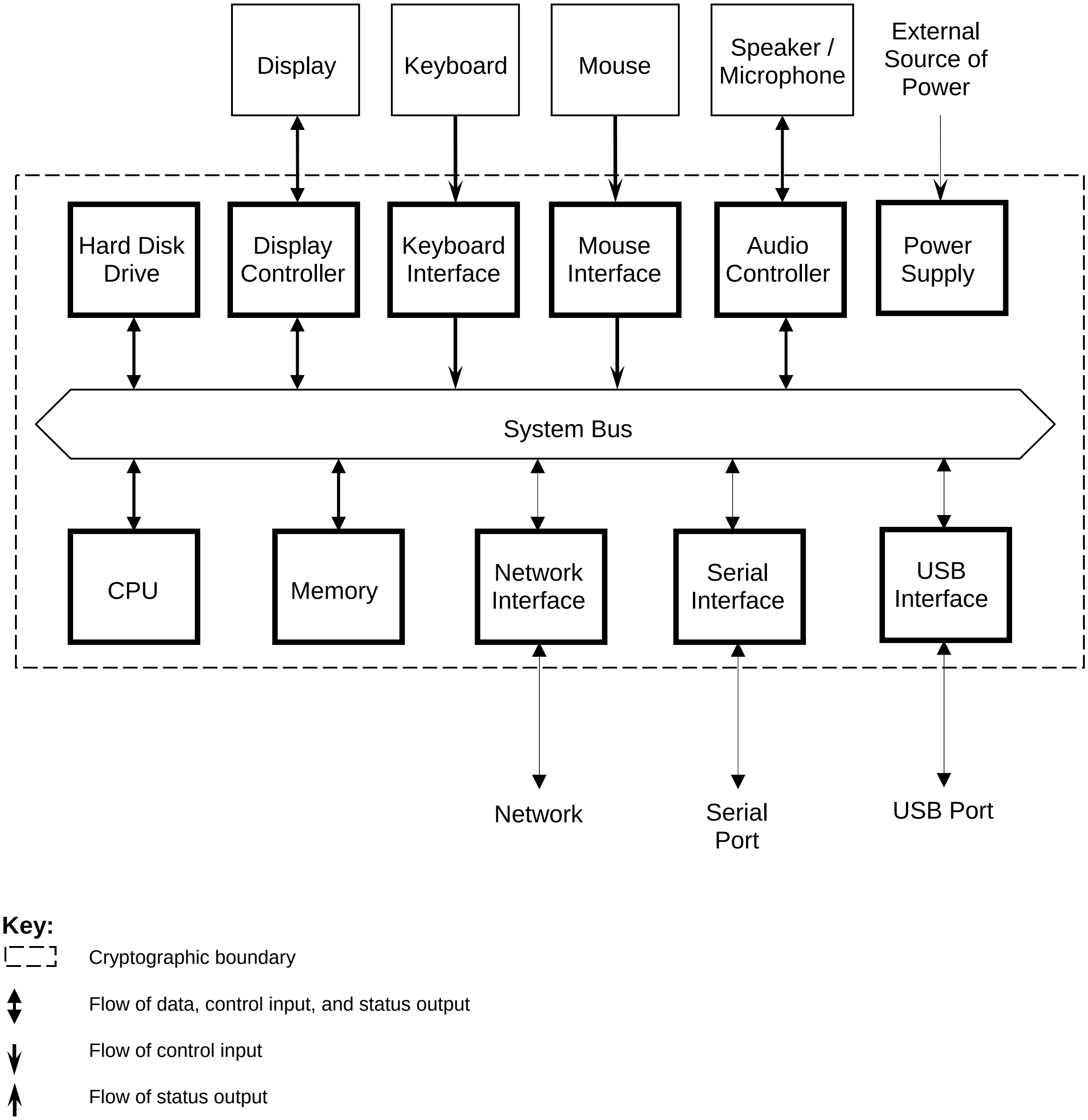
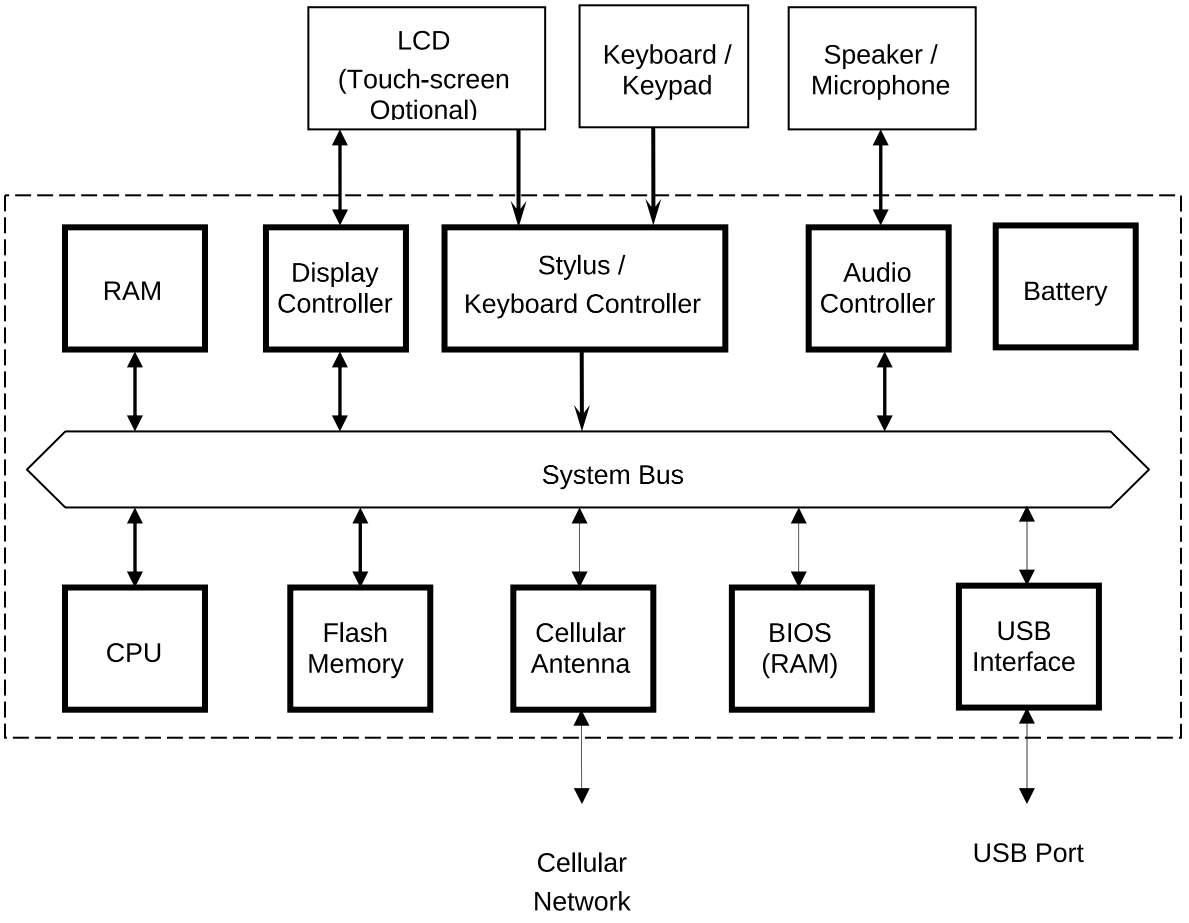


Figure 2: Cryptographic module hardware block diagram (General purpose computer)



Key:

⬚ Cryptographic boundary

↕ Flow of data, control input, and status output

↓ Flow of control input

↑ Flow of status output

Figure 3: Cryptographic module hardware block diagram (Mobile Device)

1.2 Computer hardware and OS

The module was tested and found compliant on the following computer hardware and OS:

- Ubuntu 12.04 running an Intel Xeon on a Dell PowerEdge T110
- Ubuntu 12.04 running an Intel Xeon on ESXi 5.1 on a Dell PowerEdge T110
- Ubuntu 12.04 running an AMD Opteron on a SuperMicro AS-1011S-mR2
- Ubuntu 12.04 running an AMD Opteron on ESXi 5.1 on a SuperMicro AS-1011S-mR2
- iOS v5 running on an ARMv7-based Apple A5 processor on an iPad3
- iOS v6 running on an ARMv7s-based Apple A6 processor on an iPhone5
- Android v4.1 running on an ARMv7-based Qualcomm Snapdragon processor on a Samsung Galaxy SIII.

While no claim can be made as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate, In accordance with FIPS 140-2 Implementation Guidance G5, BlackBerry further affirms that the BlackBerry Cryptographic Library for Secure Work Space will operate on any of the supported platforms and environments, including the following, while maintaining its compliance to the FIPS 140-2 Level 1 requirements:

- Android v2.2 running on an ARMv7-based Qualcomm QSD 8250 processor on an HTC Desire
- Android v2.2 running on an ARMv7-based Qualcomm QSD 8250 processor on a Dell Streak
- Microsoft Windows 7 32-bit running on an Intel Celeron (x86) processor
- uClinux 0.9.29 running on an ARMv4-based ARM 922T processor
- Fedora 14 running on an Intel Core i5 (x86) processor (with AES-NI support)
- HP-UX 11i (hpux-ia64-cc, 32 bit mode) on an Intel Itanium 2 (IA64) processor
- HP-UX 11i (hpux64-ia64-cc, 64 bit mode) on an Intel Itanium 2 (IA64) processor
- Ubuntu 10.04 running on an Intel Pentium T4200 (x86) processor
- Android 3.0 running on an ARMv7-based NVIDIA Tegra 250 T20 processor
- Linux 2.6.27 running on a PowerPC e300c3 (PPC) processor
- Microsoft Windows 7 64 bit running on an Intel Pentium 4 (x86) processor
- Ubuntu 10.04 running on an Intel :core i5 (x86) processor (with AES-NI support)
- Linux 2.6.33 running on a PowerPC32 e300 (PPC) processor
- Android 2.2 running on an ARMv7-based OMAP 3530 processor
- VxWorks 6.8 running on a TI TNETV1050 (MIPS) processor
- Linux 2.6 running on an ARMv4-based TI TMS320DM6446 processor
- Linux 2.6.32 running on an ARMv7-based TI AM3703CBP processor

- Solaris 10 32 bit running on a SPARC-T3 (SPARCv9) processor
- Solaris 10 64 bit running on a SPARC-T3 (SPARCv9) processor
- Solaris 11 32 bit running on a SPARC-T3 (SPARCv9) processor
- Solaris 11 64 bit running on a SPARC-T3 (SPARCv9) processor
- Solaris 11 32 bit running on an Intel Xeon 5675 (x86) processor
- Solaris 11 64 bit running on an Intel Xeon 5675 (x86) processor
- Solaris 11 32 bit running on an Intel Xeon 5675 (x86) processor (with AES-NI support)
- Solaris 11 64 bit running on an Intel Xeon 5675 (x86) processor (with AES-NI support)
- Oracle Linux 5 64 bit running on an Intel Xeon 5675 (x86) processor
- CascadeOS 6.1 32 bit running on an Intel Pentium T4200 (x86) processor
- CascadeOS 6.1 64 bit running on an Intel Pentium T4200 (x86) processor
- Ubuntu 10.04 32 bit running on an Intel Pentium T4200 (x86) processor
- Ubuntu 10.04 64 bit running on an Intel Pentium T4200 (x86) processor
- Oracle Linux 5 running on an Intel Xeon 5675 (x86) processor (with AES-NI support)
- Oracle Linux 6 running on an Intel Xeon 5675 (x86) processor
- Oracle Linux 6 running on an Intel Xeon 5675 (x86) processor (with AES-NI support)
- Android 4.0 running on an ARMv7-based NVIDIA Tegra 250 T20 processor
- Linux 2.6 running on a Freescale PowerPC-e500 processor
- Apple iOS 5.1 running on an ARMv7-based processor
- WinCE 6.0 running on an ARMv5- TEJ processor
- WinCE 5.0 running on an ARMv7-based processor
- Android 4.0 running on an OMAP processor
- NetBSD 5.1 runningo n a PowerPC-e500 processor
- NetBSD 5.1 running on an Intel Xeon 550 (x86) processor
- Windows 2008 32-bit under vSphere running on an Intel Xeon E3-1220v2 (x86) processor
- Windows 2008 64-bit under vSphere running on an Intel Xeon E3-1220v2 (x86) processor
- RHEL 6 32-bit under vSphere running on an Intel Xeon E3-1220v2 (x86) processor
- RHEL 6 64-bit under vSphere running on an Intel Xeon E3-1220v2 (x86) processor
- Windows 7 64-bit running on an Intel Core i5-2430M (x86) processor (with AES-NI support)
- Android 4.1 running on an ARMv7-based T1 DM3730 processor
- Android 4.1 running on an ARMv7-based T1 DM3730 processor
- Android 4.2 running on an ARMv7-based Nvidia Tegra 3 processor

- Android 4.2 running on an ARMv7-based Nvidia Tegra 3 processor
- Windows Embedded Compact 7 running on an ARMv7-based Freescale i.MX53xA processor
- Windows Embedded Compact 7 running on an ARMv7-based Freescale i.MX53xD processor
- Android 4.0 running on an ARMv7-based Qualcomm Snapdragon APQ8060 processor
- VMware Horizon Mobile 1.3 under VMware under Android 4.0 running on an ARMv7-based qualcomm MXM8X60 processor
- Apple OS X 10.7 running on an Intel Core i7-3615QM processor
- Apple iOS 5.0 running on an ARMv7-based ARM Cortex A8 processor.

1.3 Software specifications

The BlackBerry Cryptographic Library for Secure Work Space provides services in an object module format. A single source code base is used for all identified computer hardware and operating systems.

The interface into the BlackBerry Cryptographic Library for Secure Work Space 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).

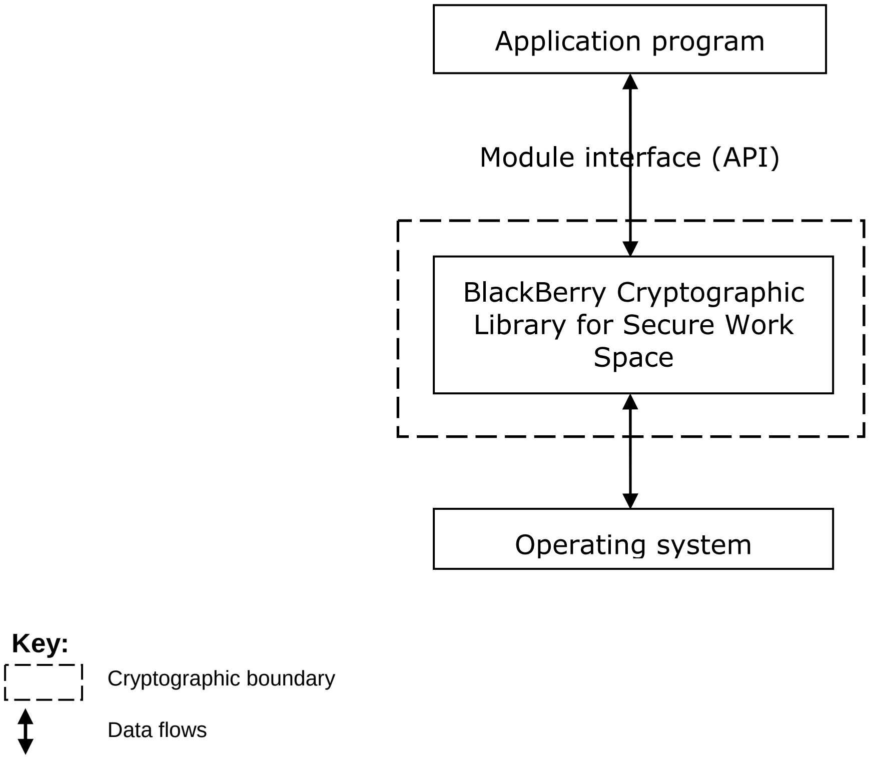


Figure 4: Cryptographic module software block diagram

2 Cryptographic Module Ports and Interfaces

The cryptographic module ports correspond to the physical ports of the GPC or mobile 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	Network port, Serial port, USB port, Cellular antenna, Bluetooth	Function calls that accept, as their arguments, data to be used or processed by the module
Data Output	Network port, Serial port, USB port, Cellular antenna, Bluetooth	Arguments for a function call that specify where the result of the function is stored
Control Input	Network port, Serial port, USB port, Cellular antenna, LCD touchscreen, Keyboard/Keypad, Power button	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output	Network port, Serial port, USB port, LEDs/LCD, Graphics controller, Audio port	Thrown exceptions for Function calls
Power Input	AC Power socket, USB port	Not applicable
Maintenance	Not supported	Not supported

3 Roles, Services, and Authentication

3.1 Roles and services

The module supports user and crypto officer roles, which may be implicitly assumed. 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, Services, CSPs and Access Types

Service	Crypto Officer	User	CSP and Type of Access
Initialization, etc.			
Initialization	X		None
Self-tests	X	X	
Show status	X	X	
Key Zeroization			
Key Zeroization	X	X	AES key – W AES CMAC Key –W Triple-DES key – W Triple-DES CMAC Key – W HMAC key – W RSA private/public key – W DSA private/public key – W ECDSA private/public key – W EC DH public/private keys – W DRBG Seed – W DRBG Entropy – W DRBG ‘C’ value – W DRBG ‘V’ value – W ANSI X9.31 RNG seed – W ANSI X9.31 seed key – W

Service	Crypto Officer	User	CSP and Type of Access
Symmetric Ciphers (AES and TDES)			
Key generation	X	X	AES key – W Triple-DES Key – W
Encrypt	X	X	AES key – RX Triple-DES key – RX
Decrypt	X	X	
Hash Algorithms and Message Authentication (SHA, HMAC,CMAC)			
Hashing	X	X	None
Message authentication	X	X	HMAC key – RX AES CMAC Key – RX Triple-DES CMAC Key – RX
Random Number Generation (pRNG)			
Instantiation	X	X	DRBG Seed – WRX DRBG Entropy – RX DRBG ‘C’ value – WRX DRBG ‘V’ value – WRX ANSI X9.31 RNG seed – WRX ANSI X9.31 seed key – RX
Seeding	X	X	
Request	X	X	
Digital Signature (DSA, ECDSA, RSA)			
Key pair generation	X	X	RSA private/public key – W DSA private/public key – W ECDSA private/public key – W
Sign		X	RSA private key – RX DSA private key – RX ECDSA private key – RX
Verify	X	X	RSA public key – RX DSA public key – RX ECDSA public key – RX

Service	Crypto Officer	User	CSP and Type of Access
Key Agreement (ECDH)			
Key Agreement	X	X	ECDH Public/Private keys – WRX
Key Wrapping (Key Transport) (RSA, AES, TDES)			
Wrap	X	X	RSA Public Key – RX AES Key – RX Triple-DES Key – RX

3.2 Security function

The BlackBerry Cryptographic Library for Secure Work Space supports many cryptographic algorithms. The set of FIPS Approved cryptographic algorithms supported by the BlackBerry Cryptographic Library for Secure Work Space is shown in Table 4.

Table 4. Approved security functions

Algorithm	Certificate Number
Symmetric Key	
AES Encryption and Decryption in ECB , CBC , CTR , CFB1 , CFB8, CFB128, and OFB modes with 128-,192-, and 256-bit key sizes	#2544
AES CCM Encryption and Decryption for 128 / 192 / 256-bit key sizes	#2544
AES GCM Encryption and Decryption for 128 / 192 / 256-bit key sizes	#2544
XTS-AES Encryption and Decryption for 128 / 256 -bit key sizes.	#2544
Triple DES: ECB, CBC, CTR, CFB1, CFB8, CFB64 and OFB modes for keying option 1 (3 keys)	#1539
Asymmetric Key	
RSA (ANSI X9.31) Key Generation with 2048 / 3072 / 4096-bit modulus; Signature Generation with 2048 / 3072 / 4096-bit modulus; Signature Verification with 1024 /1536 / 2048 / 3072 / 4096-bit modulus	#1298
RSA (PKCS #1 v1.5) Signature Generation with 2048 / 3072 / 4096-bit modulus; Signature Verification with 1024 /1536 / 2048 / 3072 / 4096-bit modulus	#1298
RSA (PSS) Signature Generation with 2048 / 3072 / 4096-bit modulus; Signature Verification with 1024 / 1536 / 2048 / 3072 / 4096-bit modulus	#1298
DSA (FIPS 186-3) Key Generation with 2048 / 3072-bit modulus	#776
DSA Signature Generation and Verification	#776
ECDSA Key Generation with All NIST Defined B, K, and P Curves	#436
ECDSA Signature Generation and Verification	#436

Hashing and Message Authentication	
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	#2145
HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512	#1565
CMAC Generate and Verify for AES and Triple DES	#2544 (AES) #1539 (TDES)
Random Number Generator	
ANSI9.31 Appendix A.2.4 PRNG	#1209
NIST SP800-90A DRBG (Hash-based)	#377
NIST SP800-90A DRBG (HMAC, no reseed)	#377
NIST SP800-90A DRBG (CTR with AES)	#377
NIST SP800-90A DRBG (Dual EC, P-256, P-384, P-521)	#377
Key Agreement Scheme	
ECDH SP800-56A, All NIST Defined B, K, and P Curves.	#89
Key Transport Scheme	
RSA encrypt/decrypt with 2048/3072/4096-bit modulus	Non-approved, but allowed

NOTE: *The following algorithms listed in the table above are considered “deprecated” or “legacy-use”. For details regarding algorithm deprecation, please refer to NIST Special Publication 800-131A.*

- *ANSI X9.31 random number generation*
- *1024-bit DSA digital signature verification*
- *1024/1536-bit RSA digital signature verification*

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)

RSA (key wrapping; key establishment methodology provides between 112 and 150 bits of encryption strength; non-compliant less than 112 bits of encryption strength)

The module includes the following non-compliant algorithms:

- 1024/1536-bit RSA key generation
- 1024/1536-bit RSA signature generation
- 1024/8192/16384-bit RSA encrypt/decrypt
- EC DH key agreement using P-192, K-163, and B163 curves

Note: By policy, the calling application is required to wrap keys with keys of equal or greater strength

Entropy:

The module generates cryptographic keys whose strengths are modified by available entropy (no assurance of the minimum strength of generated keys). It is the responsibility of the calling application to provide sufficient entropy sources to be registered as callback functions for random number generation. For the ANSI X9.31 PRNG mechanism, the callback function used must provide 128 bits of entropy. For the SP 800-90A DRBGs, the callback functions must provide the minimum amount of entropy specified in SP 800-90A Table 2 (for the Hash_DRBG and HMAC_DRBG), Table 3 (for the CTR_DRBG) and Table 4 (for the Dual_EC_DRBG). Those functions must return an error if the minimum entropy strength cannot be met.

3.3 Operator Authentication

The BlackBerry Cryptographic Library for Secure Work Space does not support authentication.

4 Physical Security

The BlackBerry Cryptographic Library for Secure Workspace is a level 1 software module, and therefore the physical requirements are not applicable for this validation.

5 Operational Environment

The BlackBerry Cryptographic Library for Secure Work Space is to run in a single-user operational environment where each user application runs in a virtually separated, independent space.

All cryptographic keys and CSPs are under the control of the host OS, which protects the keys and CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to keys and CSPs through its APIs. The module performs a Software Integrity Test using a FIPS-Approved message authentication code (HMAC SHA-1).

6 Cryptographic Key Management

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

Table 5. List of Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation /Input	Output	Storage	Zeroization	Use

AES key	AES128, 192,256 bit key
	Only AES XTS_128 (256 bits), and XTS_256 (512 bits) keys are supported.
		Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG;;
		or Input via API call parameter
			Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Encryption, decryption

AES GCM key	AES GCM 128, 192, 256 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG;
		or Input via API call parameter
			Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Encryption, decryption

AES GCM initialization vector	128 bit value	Input via API	Never	Plaintext in volatile memory	Unload module;
					API call;
					Remove Power
						Initialization vector for AES GCM

AES CMAC Key	AES 128-, 192-, or 256-bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature generation and verification

Triple-DES key	Triple-DES 192 bit key (keying option 1)	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Encryption, decryption

Triple-DES CMAC Key	Triple-DES 192 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored
				by the module
					Unload module;
					API call;
					Remove Power
						Signature generation and verification

CSP	CSP Type	Generation /Input	Output	Storage	Zeroization	Use

HMAC key	HMAC 160, 224, 256, 384, or 512 – bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Message Authentication with SHS

RSA private key	RSA, 2048, 3072, or 4096 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature generation, decapsulation

RSA public key	RSA 1024, 1536, 2048, 3072, or 4096 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature verification, encapsulation

DSA private key	DSA 224, or 256 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature generation

DSA public key	DSA 1024, 2048, or 3072 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature verification

EC DSA private key	DSA 160, 224, or 256 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature generation

EC DSA public key	DSA 1024, 2048, or 3072 bit key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Signature verification

CSP	CSP Type	Generation /Input	Output	Storage	Zeroization	Use
EC DH private key	EC DH private key agreement key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Used by host application
EC DH public key	EC DH public key agreement key	Internally generated via Approved SP 800-90A DRBG or ANSI X9.31 PRNG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Used by host application
DRBG Seed	Random data – 440 bits (SHA-1, SHA-224, SHA-256) or 880 bits (SHA-384, SHA-512)	Internally generated via approved DRBG; or Input via API call parameter	Output in plaintext via hardware appliance or mobile device INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Seeding material for SP 800-90A DRBGs

DRBG Seed	Random data – 440 or 880 bits	Generated internally using nonce along with DRBG entropy input.	Never	Keys are not persistently stored by the module	Unload module;
					API call;
					Remove Power
						Seeding material for SP 800-90A DRBGs

DRBG Entropy	256 bit value	Externally Generated;
		Input via API
			Never	Plaintext in volatile memory	Unload module;
					API call;
					Remove Power
						Entropy material for SP 800-90A DRBGs

DRBG 'C' Value	Internal state value	Internally Generated	Never	Plaintext in volatile memory	Unload module;
					API call;
					Remove Power
						Used for Hash_DRBG

DRBG 'V' Value	Internal state value	Internally Generated	Never	Plaintext in volatile memory	Unload module;
					API call;
					Remove Power
						Used for Hash_DRBG, HMAC_DRBG, and CTR_DRBG

DRBG 'Key' Value	Internal state value	Internally Generated	Never	Plaintext in volatile memory	Unload module;
					API call;
					Remove Power
						Used for HMAC_DRBG and CTR_DRBG

ANSI X9.31 Appendix A.2.4 PRNG seed	128-bit random value	Internally generated	Never exits the module	Plaintext in volatile memory	Rebooting the modules	Seeding the FIPS-Approved ANSI X9.31 PRNG
ANSI X9.31 Appendix A.2.4 PRNG key	AES 128-bit key	Internally generated	Never exits the module	Plaintext in volatile memory	Rebooting the modules	Seeding the FIPS-Approved ANSI X9.31 PRNG

7 Self-Tests

Cryptographic self-tests are performed by the module when the module is first powered up and loaded into memory as well as when a random number or asymmetric key pair is created. The following sections list the self-tests performed by the module, expected error status, and error resolution.

7.1 Power-up tests

The power-up self-tests are invoked by the FIPS_mode_set(), which invokes the fips_mode_module_set() function. If the power-up self-tests execute successfully, this function will return an integer value of “1” to the calling application to indicate success; this function will return an integer value of“0” to the calling application to indicate failure.

A failure encountered during any of the following power-up self-tests will cause the module to enter the Critical Error state, and an internal flag is set to preventing any invocation of the module’s cryptographic algorithms. The module’s running process is aborted and the Crypto Officer must unload the module from memory and reinitialize the module or the Crypto Officer must power down or restart the host system or host mobile device to clear the error state.

7.1.1 Tests upon power-up

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

The BlackBerry Cryptographic Library for Secure Work Space performs the following self-tests at power-up:

- Software integrity check (HMAC-SHA-1)
- Known Answer Tests (KATs)
  - AES-ECB KAT
  - AES-CCM KAT
  - AES-GCM KAT
  - XTS-AES KAT
  - AES CMAC KAT
  - Triple-DES KAT
  - Triple-DES CMAC KAT
  - HMAC KAT with SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512
  - RSA Digital Signature Generation KAT (2048 bit, SHA-256, PKCS#1)
  - RSA Digital Signature Verification KAT (2048 bit, SHA-256, PKCS#1)
  - DSA PCT
  - ECDSA PCT
  - ANSI X9.31 Appendix A.2.4 PRNG KAT
    - NIST SP800-90A DRBG KATs (all modes)
    - CTR_DRBG: AES, 256 bit with and without derivation function
    - HASH_DRBG: SHA256
    - HMAC_DRBG: SHA256

  - Dual_EC_DRBG: P-256 and SHA256)
- EC DH KAT.

7.1.2 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.

7.2 Conditional tests

Conditional self-tests are performed by the module whenever a new random number is generated or when a new RSA, DSA. or ECDSA key pair is generated. If any of the conditional self-tests fail, the module will log the error and halt all processing and set the internal failure flag, ensuring that data output from the module is inhibited. The CO must restart the host system or unload and reload the calling application to clear a continuous RNG test failure.

The BlackBerry Cryptographic Library for Secure Work Space performs the following conditional self-tests:

- Continuous RNG Test for the SP800-90A DRBG
- Continuous RNG Test for the ANSI X9.31 Appendix A.2.4 PRNG
- RSA pairwise consistency test for sign/verify
- RSA pairwise consistency test for wrap/unwrap
- DSA pairwise consistency test
- ECDSA pairwise consistency test.

7.3 Critical Functions Self-Tests

The BlackBerry Cryptographic Library for Secure Work Space implements the SP 800-90A Hash-based DRBG as its random number generator. This DRBG employs four critical functions which must also be tested on a regular basis to ensure the security of the SP 800-90A DRBG. If any of the critical function self-tests fail, the module will log the error and halt all system processing and set the internal failure flag, ensuring that data output from the module is inhibited. In order to resolve a cryptographic self-test error, the CO shall reset the host system of the module or unload and reload the calling application that invoked the module. Therefore, the following critical function tests are also implemented by the crypto module:

- DRBG Instantiate Critical Function Test
- DRBG Reseed Critical Function Test
- DRBG Generate Critical Function Test
- DRBG Uninstantiate Critical Function Test.

8 Mitigation of Other Attacks

The BlackBerry Cryptographic Library for Secure Work Space does not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

Appendix A Crypto Officer and User Guide

A.1 Secure Management

The Cryptographic Security Module is distributed only as part of BlackBerry's Secure Work Space software applications (BES10 MGR, and Work Connect) and is not distributed as a separate binary. The Crypto Officer is responsible for the installation of Work Connect on host systems running Android and iOS, and is responsible for the installation BES10 MGR on a host systems running Ubuntu. Successfully completing installation of the Secure Work Space software components will complete the initial set up for the Cryptographic Security Module. Running the 'show status' command on the module will, return the module's operating status and the version of the module. The value returned by the module will be version 2.0.2. This version is equivalent to version 1.0 of the BlackBerry Cryptographic Library for Secure Work Space.

A.1.1 Initialization

The module supports only an Approved mode of operation. The module requires an initialization sequence (see IG 9.5): the calling application invokes the FIPS_mode_set(), which invokes the fips_mode_module_set() function, which returns a "1" for success and "0" for failure. If the fips_mode_set() function call fails, then all cryptographic services will fail indefinitely. The calling application can call the fips_module_mode() function to determine if the module has been successfully configured for the Approved mode.

When the BES10 MGR, and Work Connect components are started, the module runs its power-up self-tests which includes software integrity test that checks the integrity of the module by using an HMAC SHA-1 digest. If the integrity check succeeds, then the module performs KATs on all the required algorithms. When the module passes all of the power-up self-tests, the module is in its FIPS-Approved mode of operation. If any self-test fails, the module enters a critical error state, ceasing all cryptographic functionality, and throws an exception to the calling application. The module must be reinstalled to leave the critical error state.

The Secure Work Space component of BES10 MGR allows the Crypto Officer to perform power-up self-tests on demand by power-cycling the host system. Crypto Officers may invoke the power-up self-tests on demand on the Secure Work Space component Work Connect by power-cycling the mobile device.

A.1.3 Management

Since the Crypto Officer cannot directly interact with the module, no specific management activities are required to ensure that the module runs securely; the module only executes in a FIPS-Approved mode of operation. If any irregular activity is noticed or the module is consistently reporting errors, then BlackBerry Customer Support should be contacted.

In the event that module power is lost and restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are re-distributed.

A.1.3 Zeroization

The module does not persistently store any keys or CSPs. All ephemeral keys used by the module are zeroized upon reboot, or session termination.

A.1.4 User Guidance

Only the module's cryptographic functionalities are available to the User. Users are responsible to use only the services that are listed in Table 5 above. In the event Module power is lost and restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are re-distributed. Although the User does not have any ability to modify the configuration of the module, they should report to the Crypto Officer if any irregular activity is noticed.

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

Acronym	Full term
PKCS	Public Key Cryptography Standard
PUB	Publication
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,* May 30, 2012.
3. NIST *Security Requirements For Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2,* August 12, 2011.
4. NIST *Security Requirements For Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2, Draft,* February 16, 2012.
5. NIST *Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2, Draft,* January 2, 2013.
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 21, 2012.
8. NIST *Frequently Asked Questions for the Cryptographic Module Validation Program,* December 4, 2007.

Document and contact information

Version	Date	Description
1.0	April 24, 2013	Document creation
1.1	February 20, 2014	Minor technical edits and updates based on SP 800-131A Transition.
1.2	March 10, 2014	Minor updates based on Lab comments

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