BlackBerry 5810 Wireless Handheld™
BlackBerry 5820 Wireless Handheld™

BlackBerry® Software Version 3.6.0.49
S/MIME Support Package Version 1.5
Hardware Version 1.0

FIPS 140-2 Non-Proprietary Security Policy
Document Version 3.3

BlackBerry® Security Team
Research In Motion

Document and Contact Information

Version	Date	Author	Description
1.0	14 February 2003	Corsec Security	Document creation.
2.0	7 March 2003	Corsec Security	
2.1	13 August 2003	Corsec Security	
3.0	28 January 2004	Dave MacFarlane	Major revisions to content and format due to comments received from CMVP.
3.1	4 February 2004	Dave MacFarlane	Clarified RSA implemented for signature generation and verification.
3.2	13 February 2004	Dave MacFarlane	Revisions due to editorial review.
3.3	10 March 2004	Dave MacFarlane	Correction to ALT-CAP-BACKSPACE key combination.

Contact	Corporate Office
BlackBerry® Security Team BlackBerrySecurity@rim.net (519) 888-7465 ext. 2921	Research In Motion Limited 175 Columbia Street West Waterloo ON Canada N2L 5Z5

Contents

Introduction 1
Cryptographic Module Specification 2
Cryptographic Module Ports and Interfaces 4
Roles, Services and Authentication 5
Physical Security 7
Cryptographic Keys and Critical Security Parameters 8
Self-Tests 10
Mitigation of Other Attacks 16
FIPS 140-2 Mode of Operation 17
BlackBerry Security Functions 17
BlackBerry® Handheld Password 17
Key Store Password 17
BlackBerry® Enterprise Server 17
Tamper Evident Seals 18
Glossary 19

List of Tables

Table 1. Module Ports and Interfaces 4
Table 2. Module Roles and Services 6
Table 3. Cryptographic Keys and CSPs 8
Table 4. Module Self-Tests 10
Table 5. IT Policy Configuration for FIPS Mode of Operation 17

Introduction

BlackBerry® is the leading wireless enterprise solution that allows users to stay connected with secure, wireless access to e-mail, corporate data, phone, web and organiser features. BlackBerry is a totally integrated package that includes hardware, software, and service, providing a complete end-to-end solution.

BlackBerry Wireless Handhelds™ support e-mail messaging and additionally support direct messaging between BlackBerry® handhelds, called PIN-to-PIN[1] messaging. By default, users can send and receive plaintext e-mail and PIN-to-PIN messages, but with the inclusion of the S/MIME software they may also send and receive S/MIME e-mail and PIN-to-PIN messages. S/MIME messages may be encrypted using AES-128, AES-192, AES-256, CAST5-128, RC2 or Triple DES and may be signed using RSA, DSA or ECDSA.

BlackBerry® handhelds protect all message types with Triple DES encryption during wireless transport. In other words, before BlackBerry® handhelds send a message it is encrypted using Triple DES, regardless of whether or not the message is already encrypted by the user using the S/MIME software.

The BlackBerry® 5800 Series of wireless handhelds are identical in functionality and both operate on Global System for Mobile communication (GSM) and General Packet Radio Service (GPRS) wireless networks, but at different frequencies. The BlackBerry 5810™ operates at 1900 MHz and targets the North American market, while the BlackBerry 5820™ operates at 900/1800 MHz and targets the European market.

The BlackBerry 5810™ and BlackBerry 5820™ are hereafter referred to as the *cryptographic module* or *module*.

[1] Each BlackBerry® handheld is uniquely identified by a personal identification number (PIN).

Cryptographic Module Specification

The plastic casing of the BlackBerry 5810™ and BlackBerry 5820™ establishes the cryptographic boundary of the module, but only the cryptographic functionality is considered herein.

The module implements the following FIPS-Approved[2] security functions:

- **AES-128**, **AES-192** and **AES-256**, as specified in FIPS PUB 197. The ECB, CBC, CFB-8, CFB-128 and OFB modes of operation are supported. The implementation has been awarded AES validation certificate # 83.
- **Triple DES**, as specified in FIPS PUB 46-3. The ECB, CBC, CFB-8, CFB-64 and OFB modes of operation are supported. The implementation has been awarded Triple DES validation certificate # 200.
- **DES**, as specified in FIPS PUB 46-3. The ECB, CBC, CFB-8, CFB-64 and OFB modes of operation are supported. The implementation has been awarded DES validation certificate # 228.
- **DSA**, as specified in FIPS PUB 186-2. The implementation has been awarded DSS validation certificate # 93.
- **ECDSA**, as specified in ANSI X9.62. The implementation is vendor affirmed to be correct.
- **RSA** (signature generation and verification), as specified in PKCS #1. The implementation is vendor affirmed to be correct.
- **RSA** (signature generation and verification), as specified in ANSI X9.31. The implementation is vendor affirmed to be correct.
- **SHA-1**, as specified in FIPS PUB 180-2. The implementation has been awarded SHS validation certificate # 147.
- **Triple DES CBC MAC**, as specified in FIPS PUB 81. In conjunction with Triple DES validation certificate # 200, the implementation is vendor affirmed to be correct.
- **DES CBC MAC**, as specified in FIPS PUB 81. In conjunction with DES validation certificate # 228, the implementation is vendor affirmed to be correct.
- **HMAC SHA-1**, as specified in FIPS PUB 198. In conjunction with SHS validation certificate # 147, the implementation is vendor affirmed to be correct.
- **Pseudo-Random Number Generator**, as specified in FIPS PUB 186-2 Appendix 3.1.

The module implements the following non-Approved security functions:

- **Skipjack**, as specified in *SKIPJACK and KEA Algorithm Specifications*, version 2.0, 29 May 1998. The ECB, CBC, CFB-8, CFB-64, OFB and X modes of operation are supported. While Skipjack is an Approved algorithm, this implementation has not been validated and is thus considered non-Approved.
- **ARC4**, as specified in *Applied Cryptography*, 1996, 2nd edition, pp. 397-398.
- **RC2**, as specified in RFC 2268.
- **RC5**, as specified in RFC 2040.

[2] A cryptographic algorithm is FIPS-Approved if it is explicitly listed in *FIPS 140-2 Annex A: Approved Security Functions for FIPS PUB 140-2*.

- **CAST5-128**, as specified in RFC 2144.
- **Rijndael**.
- **KEA**, as specified in *SKIPJACK and KEA Algorithm Specifications*, version 2.0, 29 May 1998.
- **Diffie-Hellman**, as specified in PKCS #3.
- **EC Diffie-Hellman**, as specified in IEEE P1363 Draft 13.
- **ECMQV**, as specified in IEEE P1363 Draft 13.
- **ECNR**, as specified in IEEE P1363 Draft 13.
- **ElGamal**, as specified in *Applied Cryptography*, 1996, 2nd edition, pp. 476-479.
- **SHA-256**, **SHA-384**, and **SHA-512**, as specified in FIPS PUB 180-2.
- **MD2**, as specified in RFC 1319.
- **MD4**, as specified in RFC 1320.
- **MD5**, as specified in RFC 1321.
- **RIPEMD-128**, as specified in *RIPEMD-160: A Strengthened Version of RIPEMD*, 18 April 1996.
- **RIPEMD-160**, as specified in *RIPEMD-160: A Strengthened Version of RIPEMD*, 18 April 1996.
- **CBC MAC**, as specified in FIPS PUB 81 for the following symmetric algorithms: AES, CAST5-128, RC2, RC5 and Skipjack.
- **HMAC**, as specified in FIPS PUB 198 for the following message digest algorithms: SHA-256, SHA-384, SHA-512, MD2, MD4, MD5, RIPEMD-128 and RIPEMD-160.
- **RSA OAEP**, as specified in Advances in Cryptology – Eurocrypt '94, pp. 92-111.
- **RSA PSS**, as specified in PKCS #1, version 2.1.
- **X Mode of Operation** for the AES-128, AES-192, AES-256, Triple DES and DES algorithms.

Cryptographic Module Ports and Interfaces

The following table describes the module implementation of the FIPS 140-2 ports and interfaces:

Table 1. Module Ports and Interfaces

Interface	Module Implementation
Data Input	Keyboard, serial port, internal radio modem, audio port
Data Output	Serial port, internal radio modem, audio port, liquid crystal display (LCD)
Control Input	Keyboard, trackwheel, reset button, escape button
Status Output	Light-emitting diode (LED), LCD
Power Input	Serial port

Roles, Services and Authentication

The module supports a User role and a Crypto Officer role. The module does not support a Maintenance role. Role selection is performed implicitly and is dependent on the service performed by the operator. The following services are available to the operator:

- **Inject Master Key** – Replaces the existing Master Key on the handheld with a new Master Key.
- **Configure Handheld Password** – Creates the initial handheld password or changes the existing handheld password.
- **Configure Key Store Password** – Creates the initial key store password or changes the existing key store password.
- **Reset Module** – Resets the module.
- **Power On/Off Module** – Powers the module on or off.
- **View Status** – Displays the status of the module.
- **Send E-mail Message** – Sends a plaintext or S/MIME e-mail message. Once invoked by the user, the module automatically generates a Session Key and uses it to encrypt the message. The Session Key is then encrypted with the Master Key and the encrypted message and Session Key are sent.
- **Receive E-mail Message** – Receives a plaintext or S/MIME e-mail message. This service is performed automatically by the module on behalf of the user. The module receives the encrypted message and Session Key, decrypts the Session Key with the Master Key, and then decrypts the message with the Session Key.
- **Send PIN-to-PIN Message** – Sends a plaintext or S/MIME PIN-to-PIN message. Once invoked by the user, the module automatically generates a Session Key and uses it to encrypt the message. The Session Key is then encrypted with the PIN-to-PIN Master Key and the encrypted message and Session Key are sent.
- **Receive PIN-to-PIN Message** – Receives a plaintext or S/MIME PIN-to-PIN message. This service is performed automatically by the module on behalf of the user. The module receives the encrypted message and Session Key, decrypts the Session Key with the PIN-to-PIN Master Key, and then decrypts the message with the Session Key.
- **Encrypt S/MIME Message** – Encrypts an S/MIME e-mail or PIN-to-PIN message.
- **Decrypt S/MIME Message** – Decrypts an S/MIME e-mail or PIN-to-PIN message.
- **Sign S/MIME Message** – Generates a signature of an S/MIME e-mail or PIN-to-PIN message.
- **Verify S/MIME Message** – Verifies the signature of a received S/MIME e-mail or PIN-to-PIN message.
- **Perform Self Tests** – Executes the module self-tests, as described in Self-Tests on page 10.

The following table summarises implicit role selection based on service and the associated access to critical security parameters (CSPs):

Table 2. Module Roles and Services

Service	Role Implicitly Selected	Affected Keys and CSPs	Access to Keys and CSPs
Inject Master Key	Crypto Officer	Master Key	Write
Configure Handheld Password	User	Handheld Password	Write
Configure Key Store Password	User	Key Store Password	Write
Reset Module	User	N/A	N/A
Power On/Off Module	User	N/A	N/A
View Status	User	N/A	N/A
Send E-mail Message	User	Master Key	Execute
		Session Key	Write, Execute
Receive E-mail Message	User	Master Key	Execute
		Session Key	Execute
Send PIN-to-PIN Message	User	PIN-to-PIN Master Key	Execute
		Session Key	Write, Execute
Receive PIN-to-PIN Message	User	PIN-to-PIN Master Key	Execute
		Session Key	Execute
Encrypt S/MIME Message	User	AES Key / Triple DES Key	Execute
Decrypt S/MIME Message	User	AES Key / Triple DES Key	Execute
Sign S/MIME Message	User	RSA Private Key / DSA Private Key / ECDSA Private Key	Execute
Verify S/MIME Message	User	RSA Public Key / DSA Public Key / ECDSA Public Key	Execute
Perform Self-Tests	User	N/A	N/A

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.

The operator uniquely authenticates to the module through a password mechanism. Passwords are alphanumeric and, when the module is configured as specified in FIPS 140-2 Mode of Operation on page 17, are a minimum length of five characters, making the probability of guessing the correct password less than one in 1,000,000. The module stores the SHA-1 hash of the handheld password, and the operator successfully authenticates to the module if and only if the SHA-1 hash of the supplied password matches the stored hash.

Physical Security

The module is a multi-chip standalone module contained within an opaque, hard plastic casing. Evidence of tampering is provided through the use of tamper evident seals that contain a hidden graphic. During normal use of the seals the graphic is not visible, but becomes visible when attempts are made to remove the seal. Refer to Tamper Evident Seals on page 18 for instructions on applying and using the tamper evident seals.

Cryptographic Keys and Critical Security Parameters

The following table describes the cryptographic keys, key components, and CSPs utilised by the module while in a FIPS mode of operation:

Table 3. Cryptographic Keys and CSPs

Key / CSP	Description
Handheld Password	The SHA-1 hash of the alphanumeric password used to authenticate the operator to the BlackBerry® handheld.
Key Store Password	The SHA-1 hash of the alphanumeric password used to access the handheld key store.
Master Key	A 2-key Triple DES key used to encrypt and decrypt Session Keys that are used to encrypt and decrypt e-mail messages.
PIN-to-PIN Master Key	A 2-key Triple DES key used to encrypt and decrypt Session Keys that are used to encrypt and decrypt PIN-to-PIN messages.
Session Key	A 2-key Triple DES key used to encrypt and decrypt an e-mail or PIN-to-PIN message. The module generates Session Keys using the pseudo-random number generator (PRNG) specified in FIPS PUB 186-2 Appendix 3.1.
Software Integrity Key	An RSA public key used to verify the integrity of the module software.
DES Key	A symmetric key used to encrypt and decrypt data using the DES algorithm.
Triple DES Key	A symmetric key used to encrypt and decrypt data using the Triple DES algorithm.
AES Key	A symmetric key used to encrypt and decrypt data using the AES algorithm.
DSA Key Pair	A public/private key pair used to calculate and verify digital signatures using the DSA algorithm.
ECDSA Key Pair	A public/private key pair used to calculate and verify digital signatures using the ECDSA algorithm.
RSA Key Pair	A public/private key pair used to calculate and verify digital signatures using the RSA algorithm.
DES CBC MAC Key	A DES key used to calculate and verify a message authentication code using the DES algorithm in the CBC mode of operation.
Triple DES CBC MAC Key	A Triple DES key used to calculate and verify a message authentication code using the Triple DES algorithm in the CBC mode of operation.
HMAC SHA-1 Key	A key used to calculate and verify a keyed message authentication code using the HMAC SHA-1 algorithm.

The following keys, key components and CSPs correspond to the non-Approved security functions supported by the module. Consequently, they may not be utilised while the module operates in a FIPS mode of operation:

- RC2 Key
- RC5 Key
- Skipjack Key

- CAST5-128 Key
- ARC4 Key
- Rijndael Key
- Diffie-Hellman Key Pair
- EC Diffie-Hellman Key Pair
- ElGamal Key Pair
- CBC MAC Keys for use with AES-128, AES-192, AES-256, CAST5-128, RC2, RC5 or Skipjack
- HMAC Keys for use with SHA-256, SHA-384, SHA-512, MD2, MD4, MD5, RIPEMD-128 or RIPEMD-160

Self-Tests

The following table describes the self-tests implemented by the module:

Table 4. Module Self-Tests

Test	Description
Software Integrity Test	The module implements an integrity test for the module software by verifying its 1024-bit RSA signature. The software integrity test passes if and only if the signature verifies successfully using the Software Integrity Key.
AES Known Answer Test	The module implements a known answer test (KAT) for the decrypt operation of AES-128 in the ECB mode of operation. The test passes if and only if the calculated plaintext equals the known plaintext. The AES KAT must execute successfully before the operator can access AES or Rijndael functionality.
Triple DES Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of Triple DES in the ECB mode of operation. The test passes if and only if the calculated output equals the known output for both operations. The Triple DES KAT must execute successfully before the operator can access Triple DES functionality.
Triple DES CBC Known Answer Test	The module implements a Triple DES KAT for the CBC mode of operation using known input and output. The KAT passes if and only if the calculated output equals the known output.
DES Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of DES in the ECB mode of operation. The test passes if and only if the calculated output equals the known output for both operations. The DES KAT must execute successfully before the operator can access DES functionality.
Skipjack Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of Skipjack in the ECB mode of operation. The test passes if and only if the calculated output equals the known output for both operations. The Skipjack KAT must execute successfully before the operator can access Skipjack functionality.
ARC4 Known Answer Test	The module implements a KAT for the encrypt operation of ARC4. The test passes if and only if the calculated ciphertext equals the known ciphertext. The ARC4 KAT must execute successfully before the operator can access ARC4 functionality.

Test	Description
CAST5-128 Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of CAST5-128. The test passes if and only if the calculated output equals the known output for both operations. The KAT must execute successfully before the operator can access CAST5-128 functionality.
RC2 Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of RC2. The test passes if and only if the calculated output equals the known output for both operations. The KAT must execute successfully before the operator can access RC2 functionality.
RC5 Known Answer Test	The module implements a KAT for the encrypt and decrypt operations of RC5. The test passes if and only if the calculated output equals the known output for both operations. The KAT must execute successfully before the operator can access RC5 functionality.
RSA Known Answer Test	The module implements a KAT for the encrypt operation of RSA. The test passes if and only if the calculated ciphertext equals the known ciphertext. The KAT must execute successfully before the operator can access RSA functionality.
RSA Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created RSA key pair. The RSA public key is used to encrypt a plaintext value and the resulting ciphertext is compared to the original plaintext. Next, the RSA private key is used to decrypt the ciphertext value and the resulting plaintext is compared to the original plaintext. The test passes if and only if the first comparison returns false and the second returns true. The RSA pair-wise consistency test must execute successfully in order for the key pair to be used by the operator.
ElGamal Known Answer Test	The module implements separate KATs for the encrypt and decrypt operations of ElGamal. Each test passes if and only if the calculated output equals the known output for the corresponding operation. The KAT for either operation must execute successfully before the operator can access the corresponding ElGamal functionality.
SHA-1 Known Answer Test	The module implements a SHA-1 KAT using known input and output. The KAT passes if and only if the calculated output equals the known output.
SHA-256 Known Answer Test	The module implements a KAT for SHA-256. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access the SHA-256 functionality.

Test	Description
SHA-384 Known Answer Test	The module implements a KAT for SHA-384. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access SHA-384 functionality.
SHA-512 Known Answer Test	The module implements a KAT for SHA-512. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access SHA-512 functionality.
MD2 Known Answer Test	The module implements a KAT for MD2. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access MD2 functionality.
MD4 Known Answer Test	The module implements a KAT for MD4. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access MD4 functionality.
MD5 Known Answer Test	The module implements a KAT for MD5. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access MD5 functionality.
RIPEMD-128 Known Answer Test	The module implements a KAT for RIPEMD-128. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access RIPEMD-128 functionality.
RIPEMD-160 Known Answer Test	The module implements a KAT for RIPEMD-160. The test passes if and only if the calculated message digest equals the known message digest. The KAT must execute successfully before the operator can access RIPEMD-160 functionality.
HMAC Known Answer Test	The module implements an HMAC SHA-1 KAT using known input and output. The KAT passes if and only if the calculated output equals the known output. The KAT must execute successfully before the operator can access HMAC functionality for use with any supported algorithm, such as SHA-1 or SHA-256.
CBC MAC Known Answer Test	The module implements a KAT for CBC MAC using DES. The test passes if and only if the calculated DES CBC MAC equals the known DES CBC MAC. The KAT must execute successfully before the operator can access CBC MAC functionality for use with any supported algorithm, such as DES or Triple DES.

Test	Description
Diffie-Hellman Known Answer Test	The module implements a KAT for Diffie-Hellman key agreement protocol. Using known input, a shared secret is generated. The test passes if and only if the calculated shared secret equals the known shared secret. The KAT must execute successfully before the operator can access Diffie-Hellman key agreement functionality.
Diffie-Hellman and ElGamal Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created Diffie-Hellman and ElGamal key pair. Using a known public/private key pair, the newly created key pair is used to generate a shared secret. The test passes if and only If the same shared secret is calculated with both key pairs. The Diffie-Hellman and ElGamal pair-wise consistency test must execute successfully in order for the newly created key pair to be used by the operator.
KEA Known Answer Test	The module implements a KAT for KEA protocol. Using known input, a shared secret is generated. The test passes if and only if the calculated shared secret equals the known shared secret. The KAT must execute successfully before the operator can access KEA functionality.
KEA Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created KEA key pair. Using a known public/private key pair, the newly created key pair is used to generate a shared secret. The test passes if and only If the same shared secret is calculated with both key pairs. The KEA pair-wise consistency test must execute successfully in order for the newly created key pair to be used by the operator.
DSA Known Answer Test	The module implements a KAT for DSA. Using a known key pair, the DSA signature of known data is generated and then verified. The test passes if and only if the signature verifies successfully. The DSA KAT must execute successfully before the operator can access DSA functionality.
DSA Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created DSA key pair. A DSA signature is generated and then verified. The test passes if and only if the signature verifies successfully. The DSA pair-wise consistency test must execute successfully in order for the key pair to be used by the operator.
ECDSA Known Answer Test	The module implements a KAT for ECDSA over the curve EC163K1. Using a known key pair, the signature of known data is generated and then verified. The test passes if and only if the signature verifies successfully. The ECDSA KAT must execute successfully before the operator can access any elliptic curve cryptography (ECC) functionality.

Test	Description
Elliptic Curve Cryptography Known Answer Test	The module implements a KAT for ECC over the curve specified by the operator. The supported ECC curves are EC160R1, EC163K1, EC163K2, EC163R2, EC192R1, EC224R1, EC233K1, EC233R1, EC239K1, EC256R1, EC283K1, EC283R1, EC384R1, EC409K1, EC409R1, EC521R1, EC571K1 and EC571R1. Using known input, a shared secret is generated using ECDH. The test passes if and only if the calculated shared secret equals the known shared secret. The ECC KAT must execute successfully before the operator can access any ECC functionality on the specified curve.
ECC Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created ECC key pair. An ECDSA signature is generated and then verified. The test passes if and only if the signature verifies successfully. The ECC pair-wise consistency test must execute successfully in order for the key pair to be used by the operator.
Continuous RNG Test	The module implements a continuous RNG test, as specified in FIPS 140-2, for the FIPS 186-2 Appendix 3.1 PRNG.
RNG Statistical Test	The Monobit, Poker, Runs and Long Runs statistical tests, as specified in FIPS 140-2, are executed on 20,000 bits of output generated by the FIPS PUB 186-2 Appendix 3.1 PRNG.

The module executes the following self-tests, as described in Table 4 and, during power-up without requiring operator input or action:

- Software Integrity Test
- AES KAT
- Triple DES KAT
- Triple DES CBC KAT
- DES KAT
- Skipjack KAT
- RSA KAT
- SHA-1 KAT
- SHA-256 KAT
- SHA-384 KAT
- SHA-512 KAT
- HMAC KAT
- CBC MAC KAT

- DSA KAT
- ECDSA KAT
- RNG Statistical Test

The operator may initiate the self-tests by performing a hard reset of the module by either pressing the reset button or the ALT-CAP-BACKSPACE key combination, or by power cycling the device. The Software Integrity Test is the first test executed during power up.

Mitigation of Other Attacks

The module is not designed to mitigate any specialised attacks.

FIPS 140-2 Mode of Operation

In order to operate the module in a FIPS-Approved manner, the following conditions must be met.

BlackBerry Security Functions

Only FIPS-Approved security functions may be employed while the module is operating in a FIPS mode of operation. Refer to Cryptographic Module Specification on page 2 for the list of FIPS-Approved security functions provided by the module.

BlackBerry® Handheld Password

During initial configuration of the BlackBerry® handheld, the operator must configure a handheld password by performing the following steps:

1. Select the **Options** icon in the home screen.
2. Select **Security** in the **Options** screen.
3. Set the **Password** option to *Enabled*, if it is currently set to *Disabled*.
4. Set the **Password Timeout** option to a value no greater than 10 minutes.
5. Click the thumbwheel to display the **Security** menu and select **Save**.
6. Enter the handheld password in the **New Password** dialog that appears.
7. Re-enter the handheld password in the **Verify New Password** dialog that appears.

Key Store Password

During initial configuration of the BlackBerry® handheld, the operator must configure a key store password by performing the following steps:

1. Select the **Options** icon in the home screen.
2. Select **Handheld Key Store** in the **Options** screen.
3. Enter the key store password in the **New Key Store Password** dialog that appears.
4. Re-enter the key store password in the **Verify Key Store Password** dialog that appears.

BlackBerry® Enterprise Server

The following IT Policies must be configured on the BlackBerry Enterprise Server that serves the module, as shown:

Table 5. IT Policy Configuration for FIPS Mode of Operation

IT Policy	Value	Description
Password Required	True	Forces the operator to use a handheld password.
Minimum Password Length	5	Sets a minimum length on the handheld password.
User Can Disable Password	False	Disallows the operator from disabling the handheld password.
Enable WAP Config.	False	Disables the WAP browser.

IT Policy	Value	Description
Suppress Password Echo	True	Prevents the password characters from being displayed when the operator is attempting to authenticate to the handheld.
Disallow Third Party Application Downloads	True	Prevents additional software from being loaded onto the handheld.
Allow External Connections	False	Prevents external connections from the handheld.
TLS Restrict FIPS Ciphers	True	Forces the use of FIPS-Approved algorithms when using TLS.
SMIME Require FIPS Ciphers	True	Forces the use of FIPS-Approved algorithms when creating and sending S/MIME e-mail or PIN-to-PIN messages.

Tamper Evident Seals

For the purposes of applying the tamper-evident seals, the front of the BlackBerry® handheld is the side of the device with the LCD and keyboard. The tamper-evident seals must be applied, as follows:

1. Ensure the ambient air temperature is above 10°C.
2. Clean the enclosure of any grease, dirt, or oil. Alcohol-based cleaning pads are recommended for this purpose.
3. Select an exposed screw on the back of the BlackBerry® handheld.
4. Place a tamper-evident seal over the selected screw so that it is no longer exposed.
5. Repeat steps 3 and 4.
6. Allow one hour for the adhesive on the tamper-evident seal to properly cure.

The tamper evident seals should be examined periodically for the following signs of tampering:

- Seal is missing
- Seal appearance is distorted
- Hidden graphic is visible

Glossary

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
ARC	Alleged Rivest Cipher
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CSP	critical security parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EC	elliptic curve
ECB	Electronic Codebook
ECC	elliptic curve cryptography
ECMQV	elliptic curve Menezes-Qu-Vanstone
ECNR	elliptic curve Nyberg-Rueppel
FIPS	Federal Information Processing Standard
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
HMAC	Keyed-Hashed Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers
KAT	known answer test
KEA	Key Exchange Algorithm
LCD	liquid crystal display
LED	light emitting diode
MAC	Message Authentication Code
MD	Message Digest
OAEP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
PIN	personal identification number
PKCS	Public Key Cryptography Standard
PRNG	pseudo-random number generator
RC	Rivest Cipher
RFC	Request For Comment
RIM	Research In Motion

RIPE	Race Integrity Primitives Evaluation
RNG	random number generator
RSA	Rivest, Shamir, Adleman
S/MIME	Secure Multipurpose Internet Mail Extension
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
TLS	Transport Layer Security
WAP	Wireless Application Protocol