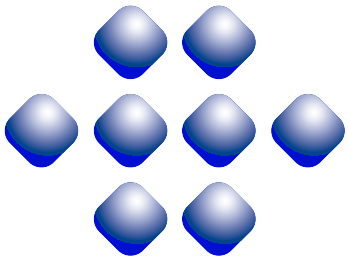
Security Builder® FIPS Module

Versions 4.0 B and 4.0 S

FIPS 140-2 Non-Proprietary Security Policy

Certicom Corp.

February 18, 2008



certicom™

Contents

1 Introduction

1.1 Overview

This is a non-proprietary Federal Information Processing Standard (FIPS) 140-2 Security Policy for Certicom's **Security Builder® FIPS Module Versions 4.0 B and 4.0 S** (SB FIPS Module). SB FIPS Module is a cryptographic toolkit for C language users, providing services of various cryptographic algorithms such as hash algorithms, encryption schemes, message authentication, and public key cryptography. This Security Policy specifies the rules under which SB FIPS Module must operate. These security rules are derived from the requirements of FIPS 140-2 [1], and related documents [6, 7, 8].

1.2 Purpose

This Security Policy is created for the following purposes:

1. It is required for FIPS 140-2 validation.
2. To outline SB FIPS Module's conformance to FIPS 140-2 Level 1 Security Requirements.
3. To provide users with how to configure and operate the cryptographic module in order to comply with FIPS 140-2.

1.3 References

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 19, 2007.

[3] NIST *Security Requirements For Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2*, November 4, 2004.

[4] NIST *Security Requirements For Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2*, March 19, 2007.

[5] NIST *Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2*, March 19, 2007.

[6] NIST *Derived Test Requirements for FIPS 140-2*, Draft, March 24, 2004.

[7] NIST *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*, March 19, 2007.

[8] NIST *Frequently Asked Questions for the Cryptographic Module Validation Program*, December 8, 2006.

1.4 Change Notes

The following are placed here by RCS upon check-in.

```
$Log: FIPSModuleHandHeldSecurityPolicy.tex,v $
Revision 1.3.14.14  2008/02/18 05:39:19  ayamada
Corrections and clarifications based on the comments from CMVP.

Revision 1.3.14.13  2008/01/10 19:29:01  ayamada
1. Correction on the firmware cryptographic boundary.
2. Correction: Software -> Firmware.

Revision 1.3.14.12  2008/01/10 16:24:50  ayamada
1. Correction in Figure 3: Operating System -> Firmware Image
2. Additions to Table 3: Initialization and Deinitialization.
3. Additions to Tables 3 and 5: Zeroization (i.e., destruction).

Revision 1.3.14.11  2008/01/04 14:11:41  ayamada
Added a table on keys and CSPs.
A typo fix as well.

Revision 1.3.14.10  2007/07/03 11:38:48  ayamada
Correction on Figure 3.

Revision 1.3.14.9  2007/06/28 15:18:35  ayamada
Editorial correction.

Revision 1.3.14.8  2007/06/28 14:52:07  ayamada
Further clarification on network port.

Revision 1.3.14.7  2007/06/27 14:07:49  ayamada
More correction.

Revision 1.3.14.6  2007/06/26 18:28:56  ayamada
Some editorial corrections.

Revision 1.3.14.5  2007/06/26 12:45:58  ayamada
Added the algorithm certificate numbers for the Scanner.

Revision 1.3.14.4  2007/06/06 18:55:50  ayamada
Added the module for the scanner.

Revision 1.3.14.3  2007/05/03 12:40:10  ayamada
Added further information on the hardare and firmware.

Revision 1.3.14.2  2007/05/02 19:11:20  ayamada
Correction on the haedware descriptions.

Revision 1.3.14.1  2007/04/26 13:33:45  ayamada
Brought in the latest version from the trunk.

Revision 1.10  2007/04/19 13:56:28  ayamada
More accurate description of the hardware.

Revision 1.9  2007/04/19 13:27:59  ayamada
Correction in the instruction in Appendix.

Revision 1.8  2007/04/19 13:23:22  ayamada
Clarification in the Appendix and typo fix.

Revision 1.7  2007/04/09 16:06:22  ayamada
Added more on the Operational Environment.

Revision 1.6  2007/04/09 15:18:59  ayamada
Editorial correction.

Revision 1.5  2007/04/03 15:45:34  ayamada
```

Correction on the HW diagram and description.

Revision 1.4 2007/04/03 14:59:27 ayamada
Hardware and firmware information is added.

Revision 1.3 2007/03/22 18:20:09 ayamada
Some corrections.

Revision 1.2 2007/03/22 12:46:35 ayamada
Typo fix.

Revision 1.1 2007/03/21 19:46:10 ayamada
Initial revision.

2 Cryptographic Module Specification

SB FIPS Module is a multiple-chip standalone firmware cryptographic module.

2.1 Physical Specifications

2.1.1 Version 4.0 B Hardware

The hardware component of Hand Held Products BASE 20205B-FIPSE consists of the following devices:

1. CPU (ARM 920T)
2. Memory
   (a) Working memory is located on the RAM containing 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. LED
4. Serial Port
5. Wireless Network Interface
6. Power Supply

The configuration of this component is illustrated in Figure 1.

2.1.2 Version 4.0 S Hardware

The hardware component of Hand Held Products Scanner 4820SF-FIPSE consists of the following devices:

1. CPU (ARM 920T)
2. Memory
   (a) Working memory is located on the RAM containing 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.

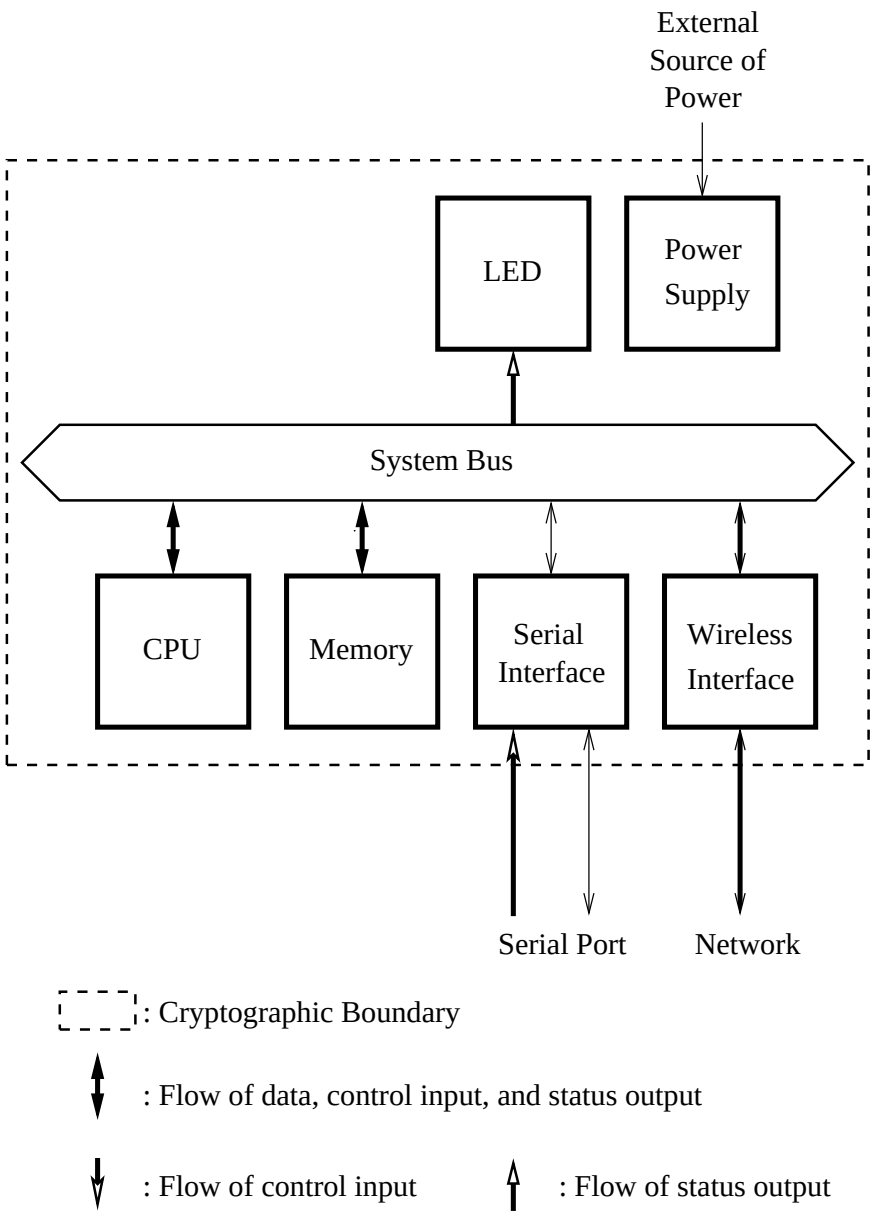


Figure 1: Cryptographic Module 4.0 B Hardware Block Diagram

3. Scanner
4. Button
5. LED
6. Serial Port
7. Wireless Network Interface
8. Battery

The configuration of this component is illustrated in Figure 2.

2.2 Firmware Specifications

SB FIPS Module is manufactured by Certicom Corp., providing services to the C computer language users.

The interface into SB FIPS Module is via 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).

2.2.1 Version 4.0 B Firmware

SB FIPS Module 4.0 B is to be linked with the Hand Held Products BASE firmware 31205423-052.

2.2.2 Version 4.0 S Firmware

SB FIPS Module 4.0 S is to be linked with Hand Held Products Scanner firmware 31205480-025.

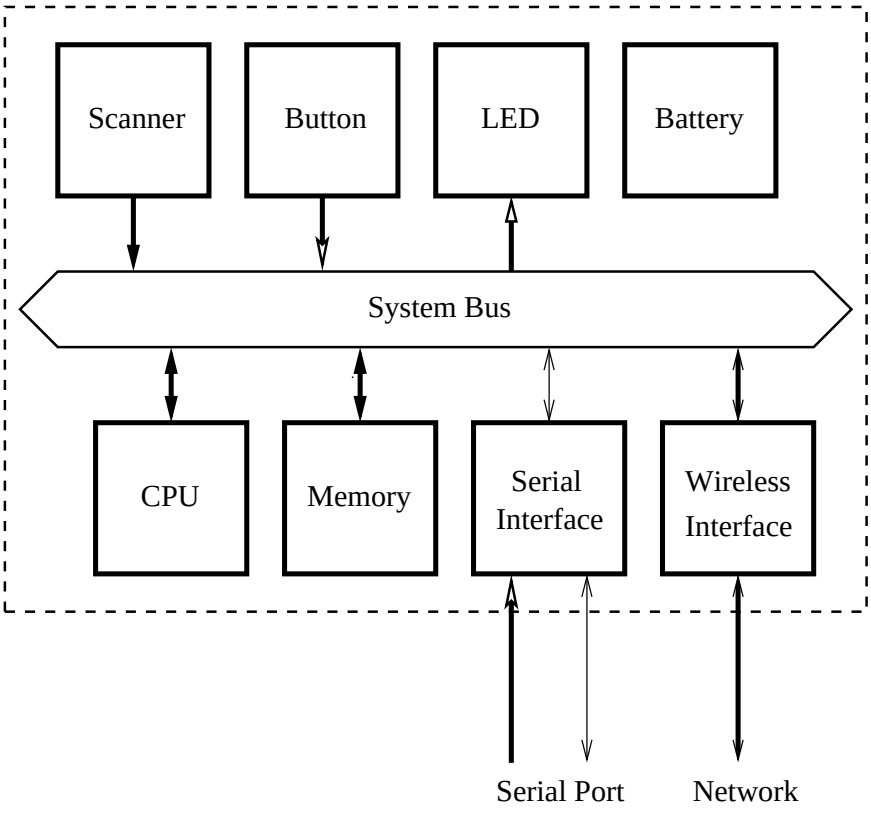


Figure 2: Cryptographic Module 4.0 S Hardware Block Diagram

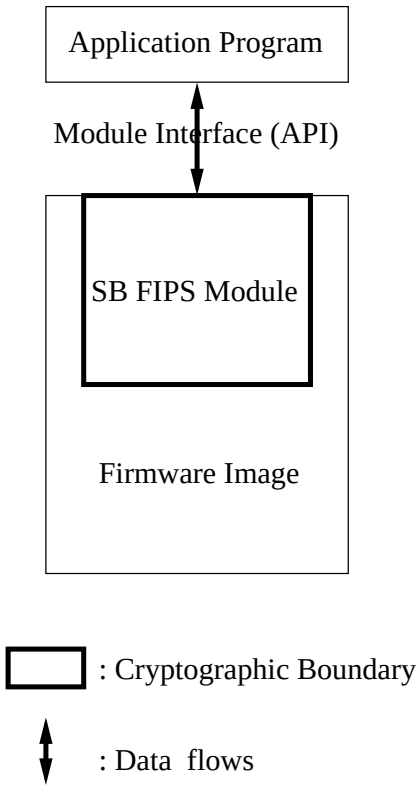


Figure 3: Cryptographic Module Firmware Block Diagram

3 Cryptographic Module Ports and Interfaces

3.1 Version 4.0 B Ports and Interfaces

The physical and logical interfaces for SB FIPS Module 4.0 B are summarized in Table 1.

Table 1: Version 4.0 B Logical and Physical Interfaces

I/O	Logical Interface	Physical Interface
Data Input	API	Serial port
Data Output	API	Serial/Wireless port
Control Input	API	Serial Port
Status Output	Return Code	LED
Power Input	Initialization Function	The power supply is the power interface.
Maintenance	Not supported	Not supported

3.2 Version 4.0 S Ports and Interfaces

The physical and logical interfaces for SB FIPS Module 4.0 S are summarized in Table 2.

Table 2: Version 4.0 S Logical and Physical Interfaces

I/O	Logical Interface	Physical Interface
Data Input	API	Scanner/Serial port
Data Output	API	Serial/Wireless port
Control Input	API	Button/Serial port
Status Output	Return Code	LED
Power Input	Initialization Function	N.A. (Battery is included.)
Maintenance	Not supported	Not supported

4 Roles, Services, and Authentication

4.1 Roles

SB FIPS Module supports Crypto Officer and User Roles. These roles are enforced by this Security Policy. The Crypto Officer has the responsibility for installing SB FIPS Module (see Table 3).

Table 3: Roles and Services

Service	Crypto Officer	User
Installation, etc.		
Installation	×	
Uninstallation	×	
Initialization	×	×
Deinitialization	×	×
Self-tests	×	×
Show status	×	×
Keys and CSPs Zeroization	×	×
Symmetric Cipher (AES)		
Key generation	×	×
Encrypt	×	×
Decrypt	×	×
Hash Algorithms and Message Authentication (SHA, HMAC)		
Hashing	×	×
Message Authentication	×	×
Random Number Generation (pRNG)		
Instantiation	×	×
Seeding	×	×
Request	×	×
Digital Signature (DSA)		
Key pair generation	×	×
Sign	×	×
Verify	×	×
Key Agreement (DH)		
Key pair generation	×	×
Shared secret generation	×	×

In order to operate the module securely, it is the Crypto Officer and User's responsibility to confine calls to those methods that have been FIPS 140-2 Approved or allowed. Thus, in the approved mode of operation, all Roles shall confine themselves to calling FIPS Approved or allowed algorithms, as marked in Table 4.

4.2 Services

SB FIPS Module supports many cryptographic algorithms. The set of cryptographic algorithms supported by SB FIPS Module are given in Table 4.

Table 4: Supported Algorithms and Standards

	Algorithm	**FIPS Approved or allowed**	**Cert Number**	
			4.0 B	**4.0 S**
Block Ciphers	AES (ECB, CBC, CFB128, OFB128, CTR) [FIPS 197]	×	#547	#590
Hash Functions	SHA-1 [FIPS 180-2]	×	#612	#641
	SHA-224 [FIPS 180-2]	×	#612	#641
	SHA-256 [FIPS 180-2]	×	#612	#641
Message Authentication	HMAC-SHA-1 [FIPS 198]	×	#288	#307
	HMAC-SHA-224 [FIPS 198]	×	#288	#307
	HMAC-SHA-256 [FIPS 198]	×	#288	#307
RNG	ANSI X9.62 RNG [ANSI X9.62]	×	#315	#336
Digital Signature	DSA [FIPS 186-2]	×	#222	#232
Key Agreement	DH [ANSI X9.42]	×		

The AES, SHA-1, SHA-224, SHA-256, HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA256, RNG, and DSA algorithms have been validated to comply with FIPS. SB FIPS Module also supports a FIPS allowed key establishment technique (key agreement), DH. In order to operate the module in compliance with FIPS, only these FIPS Approved or allowed algorithms should be used.

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

Table 5: Key and CSP, Key Size, Security Strength, and Access

Algorithm	**Key and CSP**	Key Size	Strength	**Access**
AES	key	128-256 bits	128-256 bits	Create, Read, Use, Destroy
HMAC	key	160-256 bits	80-128 bits	Create, Read, Use, Destroy
pRNG	seed key, seed	160 bits	80 bits	Use
DSA	key pair	1024-15360 bits	80-256 bits	Create, Read, Use, Destroy
DH	static/ephemeral key pair	1024-15360 bits	80-256 bits	Create, Read, Use, Destroy

4.3 Operator Authentication

SB FIPS Module does not deploy authentication mechanism. The roles of Crypto Officer and User are implicitly selected by the operator.

5 Finite State Model

The Finite State model contains the following states:

- Installed/Uninitialized
- Initialized
- Self-Test
- Idle
- Crypto Officer/User
- Error

The following is 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, i.e., the module is loaded on the memory, turning to the Initialization state. Then, it transits to the Self-Test state automatically, running the Power-up Tests. While in the Self-Test state, all data output via the data output interface is prohibited. On success the module enters Idle; on failure the module enters Error and the module is disabled. From the Error state the Crypto Officer may need to re-install to attempt correction.
3. From the Idle state (which is only entered 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 Error and the module is disabled.
6. When On-demand Self-test is executed, the module enters the Self-Test state. On success the module enters Idle; on failure the module enters Error and the module is disabled.
7. When the de-initialization command is executed, the module goes back to the Installed/Uninitialized state.

6 Physical Security

SB FIPS Module operates on a device where a production grade enclosure is used.

7 Operational Environment

SB FIPS Module runs in the non-modifiable environment, where the device is a base station for a hand held scanner.

8 Cryptographic Key Management

SB FIPS Module provides the underlying functions to support FIPS 140-2 Level 1 key management. The user will select FIPS Approved or allowed algorithms and will handle keys with appropriate care to build up a system that complies with FIPS 140-2. It is the Crypto Officer and User's responsibility to select FIPS 140-2 validated algorithms (see Table 4).

8.1 Key Generation

SB FIPS Module provides FIPS 140-2 compliant key generation. The underlying random number generation uses a FIPS Approved method, the ANSI X9.62 RNG [4].

8.2 Key Establishment

SB FIPS Module provides the following FIPS allowed key establishment technique [5]:

1. Diffie-Hellman (DH)

The DH key agreement technique implementation supports modulus sizes from 512 bits to 15360 bits that provides between 56 and 256 bits of security strength, where 1024 bits and above must be used to provide minimum of 80 bits of security.

It is responsibility of the application to ensure that the appropriate key establishment techniques are applied to the appropriate keys.

8.3 Key Entry and Output

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

8.4 Key Storage

SB FIPS Module does not store keys.

8.5 Zeroization of Keys

SB FIPS Module functions zeroize all intermediate security sensitive material. All CSPs are zeroized when they are no longer needed by calling destroy functions. Destruction of CSP is enforced in a manner such that missed destruction will make SB FIPS Module no longer functional.

9 Self-Tests

9.1 Power-up Tests

9.1.1 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 AES, SHS, HMAC-SHS, and RNG. For DSA, Pair-wise Consistency Test is used.

2. **Firmware Integrity Test:**
   The firmware integrity test deploys HMAC-SHA-256 to verify the integrity of the module.

9.1.2 On-Demand Self-Tests

On-demand self tests may be invoked by the Cryptographic Officer or User by invoking a function, which is described in the Crypto Officer And User Guide in Appendix A.

9.2 Conditional Tests

The Continuous RNG Test is executed on all RNG generated data, examining the first 160 bits of each requested random generation for repetition. This ensures that the RNG is not stuck at any constant value.

Also, upon each generation of a DSA 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.

9.3 Failure of Self-Tests

Failure of the Self-tests places the cryptographic module in the Error state, wherein no cryptographic operations can be performed. It is a hard error, and re-loading, and possibly re-building as well, of the firmware image is necessary to attempt recovery.

10 Design Assurance

10.1 Configuration Management

A configuration management system for the cryptographic module is employed and has been described in a document to the testing laboratory. It uses the Concurrent Versioning System (CVS) to track the configurations.

10.2 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.

10.3 Development

Detailed design information and procedures have been described in documentation submitted to the testing laboratory. The source code is fully annotated with comments, and is also submitted to the testing laboratory.

10.4 Guidance Documents

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

11 Mitigation of Other Attacks

SB FIPS Module implements mitigation of the following attacks:

1. Attack on biased private key of DSA

11.1 Attack on Biased Private Key of DSA

The standards for choosing ephemeral values in DSA 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 which 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`

A Crypto Officer And User Guide

A.1 Installation

In order to carry out a secure installation of SB FIPS Module, the Crypto Officer must follow the procedure described in this section.

A.1.1 Installing

The Crypto Officer is responsible for the installation of SB FIPS Module. Only the Crypto Officer is allowed to install the product.

Build the firmware image to be loaded by linking the object module, `sbgse4.o`, to the application. Then load the image that includes the object module to the device.

A.1.2 Uninstalling

Overwrite the object module, `sbgse4.o`, on the device.

A.2 Commands

A.2.1 Initialization

`sbg4_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` will be returned and the module will be enabled.

A.2.2 De-initialization

`sbg4_FIPS140Deinitialize()`

This function de-initializes the module.

A.2.3 Self-Tests

`sbg4_FIPS140RunTest()`

This function runs a series of self-tests, and return `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 will be disabled. Section A.3 of this document describes how to recover from the disabled state.

A.2.4 Show Status

`sbg4_FIPS140GetState()`

This function will return the current state of the module.

A.3 When Module is Disabled

When SB FIPS Module becomes disabled, attempt to bring the module back to the Installed state by calling `sbg4_FIPS140Deinitialize()`, and then to initialize the module using `sbg4_FIPS140Initialize()`. If the initialization is successful, the module is recovered. If this attempt fails, uninstall the module and re-install it. If the module is initialized successfully by this re-installation, the recovery is successful. If this recovery attempt fails, it indicates a fatal error. Please contact Certicom Support immediately.