Security Builder® FIPS Module

Version 6.3.0

FIPS 140-2 Non-Proprietary
Security Policy

Certicom Corp.

February 1, 2019

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 Version 6.3.0** (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 information on 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*, Draft, May 10, 2017.

[3] NIST *Security Requirements for Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2*, Draft, December 21, 2016.

[4] NIST *Security Requirements for Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2*, Draft, January 04, 2016.

[5] NIST *Security Requirements for Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2*, Draft, May 10, 2017.

[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,* May 10, 2017.

[8] NIST *Frequently Asked Questions for the Cryptographic Module Validation Program*, October 28, 2016.

1.4 Change History

Change history is recorded in Table 1.

Table 1: Change History

Revision	Date	Author	Description
0.1	2017/07/10	H.W.	Initial revision. Created based on the Security Policy for SB FIPS Module 6.0.
0.2	2017/10/31	H.W.	Updated after initial review.
0.3	2018/02/05	R.T.	Editorial corrections.
0.4	2018/03/05	H.W.	Address Lab Observations.
0.5	2018/03/20	H.W.	Address Lab Observations.
0.6	2018/04/10	H.W.	Address Lab Observations.
0.7	2018/04/25	H.W.	Address Lab Observations.
0.8	2018/05/03	H.W.	Address Lab Observations.
0.9	2018/08/08	H.W.	Address CMVP comments.
0.10	2018//08/20	R.T.	Address CMVP comments.
0.11	2018/11/28	R.T.	Address CMVP comments.
0.12	2018/12/13	R.T.	Address CMVP comments.
0.13	2018/12/18	R.T.	Address further CMVP comments.
0.14	2019/02/01	R.T.	Address further CMVP comment.

2 Cryptographic Module Specification

SB FIPS Module is a multiple-chip standalone software cryptographic module in the form of an object that operates with the following components:

- Commercially available general-purpose computer hardware.
- Commercially available Operating System (OS) that runs on the computer hardware.

Table 2: level of validation

FIPS 140-2 Section Title	**Validation 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
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	1

2.1 Physical Specifications

The tested platforms consist of the following devices:

1. CPU (Microprocessor)
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

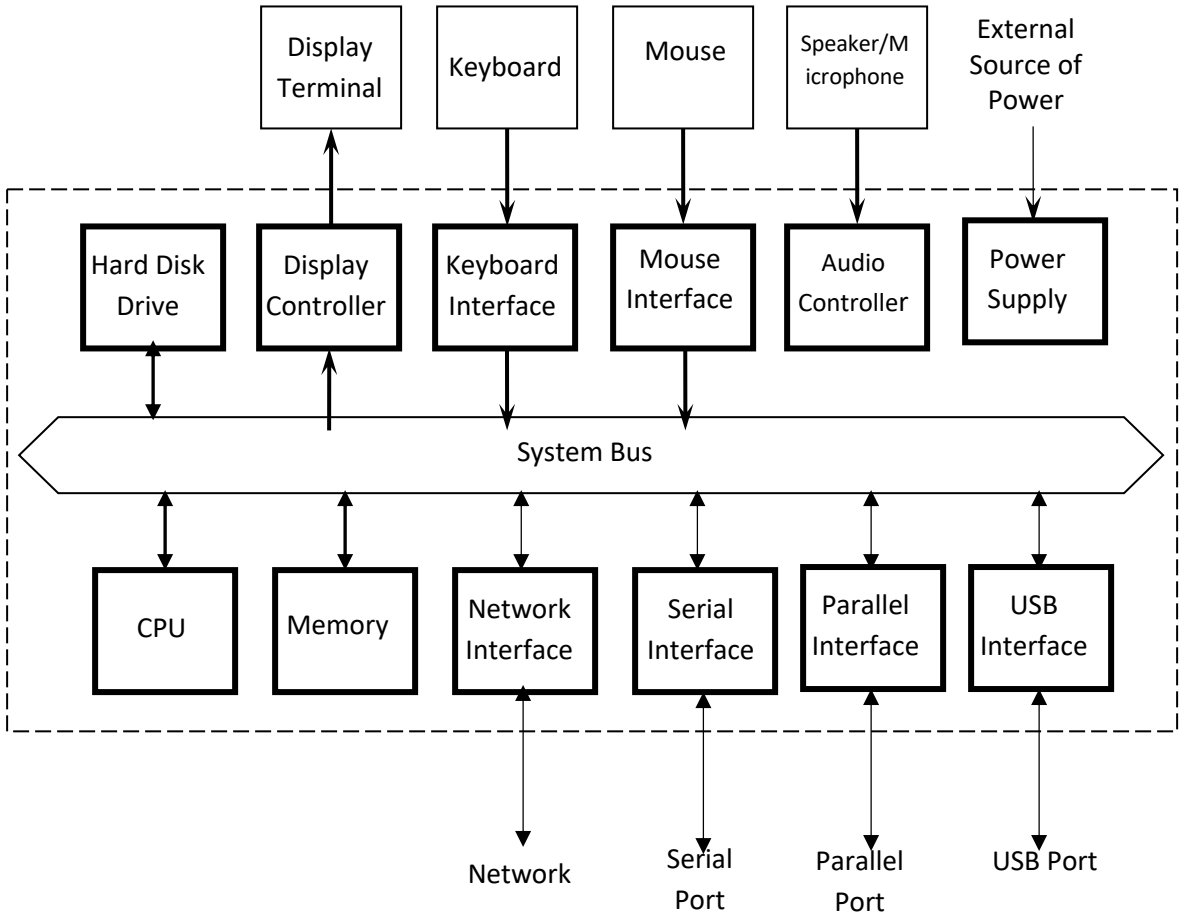
    Key storage is not deployed in this module.

    (b) Program memory is also located on RAM.
3. Hard Disk (or disks)
4. Display Controller, including Touch Screen Controller
5. Keyboard Interface
6. Mouse Interface, including Trackball Interface
7. Audio Controller
8. Network Interface
9. Serial Interface
10. Parallel Interface

11. USB Interface
12. Power Supply

The configuration of this component is illustrated in Figure 1.

Figure 1: Cryptographic Module Hardware Block Diagram



: Physical Cryptographic Boundary

: Flow of data, control input, and status output

: Flow of control input

: Flow of status output

2.2 Computer Hardware and OS

The combinations of computer hardware and OS include the following representative platforms.

1. QNX SDP 7 on Renesas R-Car Pro(M3) development board, Cortex-A57/A-53 64-bit ARMv8 processor.
2. QNX SDP 7on BeagleBone Black development board, TI AM335x ARMv7 processor.
3. QNX SDP 7 on SUPERMICRO A2SDi-4C-HLN4F-O server mother board, Intel Atom C3558 64-bit x86 processor.
4. QNX SDP 6.6 on Boundary Devices NITROGEN6X i.MX6Q development board, Cortex-A8 ARMv7 processor.
5. Android 7.1.2 on BlackBerry KEYone smart phone, Qualcomm MSM8953 64-bit ARMv8 processor

For the platforms listed above, SB FIPS Module is tested supporting NEON instructions on ARMv7 and ARMv8 processor, supporting Crypto Extensions instructions on ARMv8 processor, supporting AES-NI instructions on 64-bit x86 processor. It has also been tested that when these instruction sets are missing or turned off, SB FIPS Module still works.

SB FIPS Module is also suitable for any platforms of any manufactures with compatible processors and equivalent system configurations, and compatible OS versions. For example, an identical SB FIPS Module can be used on any compatible QNX for 64-bit x86 processors, or Android for ARMv8 processors. SB FIPS Module will run on such platforms and OS versions while maintaining its compliance to the FIPS 140-2 Level 1 requirements.

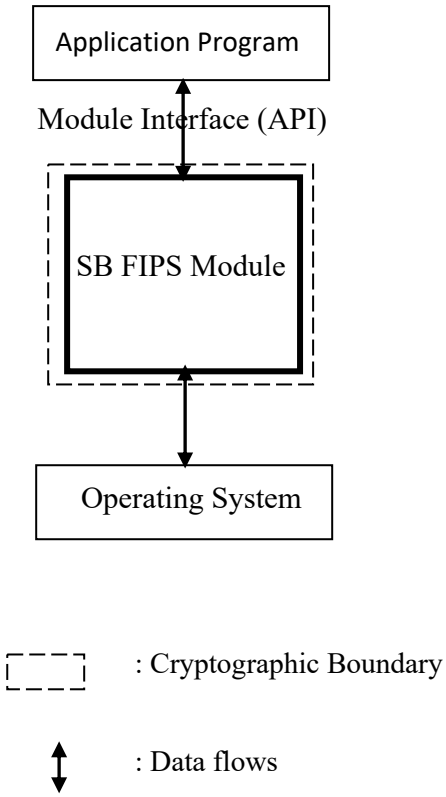
Additionally, the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate.

2.3 Software Specifications

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

The interface into SB FIPS Module is via Application Programmer's Interface (SB 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 1).

Figure 2: Cryptographic Module Software Block Diagram



3 Cryptographic Module Ports and Interfaces

The physical and logical interfaces are summarized in Table 3.

Table 3: Logical and Physical Interfaces

I/O	Logical Interface	Physical Interface
Data Input	API	Keyboard, Mouse, USB ports, Serial, and Ethernet port
Data Output	API	USB ports, Serial ports and Ethernet port
Control Input	API	Keyboard and Mouse
Status Output	API return code and state returned by sbg_FIPS140GetState()	Display
Power Input	OS API to load the module (library)	The Power Supply is the power interface.
Maintenance	Not supported	Not supported

The APIs logically separate data input from control input, and data output and status output.

4 Roles, Services, and Authentication

4.1 Roles and Services

SB FIPS Module supports Crypto Officer and User Roles (see Table 4). These roles are enforced by this Security Policy.

Table 4: Roles and Services

Service	Roles	Key/CSP	Permission
Installation, etc.			
Installation	Crypto Officer/User	None	execute
Self-tests	Crypto Officer/User	All Keys/CSPs listed in Table 6, with the exception of ephemeral key pair generation.	execute
Show status	Crypto Officer/User	None	execute
Symmetric Ciphers (AES and Triple-DES)			
Key generation	User	AES Key, Triple-DES key	read/write/execute
Key input	User	AES Key, Triple-DES key	read/write/execute
Encrypt	User	AES Key, Triple-DES key	read/write/execute
Decrypt	User	AES Key, Triple-DES key	read/write/execute
Key zeroization	User	AES Key, Triple-DES key	read/write/execute
Hash Algorithms and Message Authentication (SHA, HMAC)			
Hashing	User	None	read/write/execute
Message Authentication	User	HMAC Key	read/write/execute
Seed from system (NDRNG)			
Request	User	None	read/write/execute
Random Number Generation (DRBG)			
Instantiation	User	Entropy input, V, Key or C	read/write/execute
Seeding	User	Entropy input, V, Key or C	read/write/execute
Request	User	Entropy input, V, Key or C	read/write/execute
CSP/Key zeroization	User	Entropy input, V, Key or C	read/write/execute
Digital Signature (DSA, ECDSA, RSA)			
Key pair generation	User	Key pair	read/write/execute
Key pair input	User	Key pair	read/write/execute
Sign	User	Private Key	read/write/execute
Verify	User	Public Key	
Key zeroization	User	Key pair	read/write/execute
Key Establishment (DH, ECDH, ECMQV, RSA)			
Key pair generation	User	Key pair	read/write/execute
Key pair input	User	Key pair	read/write/execute
Shared secret generation (DH, ECDH, and ECMQV)	User	Public Key and Private Key, secret	read/write/execute
Wrap and Unwrap (RSADP)	User	RSA Public Key and RSA Private Key	read/write/execute
Wrap and Unwrap (RSA-OAEP)	User	RSA Public Key and RSA Private Key	read/write/execute
Key zeroization	User	Key pair	read/write/execute

SB FIPS Module does not support multiple concurrent operators.

SB FIPS Module supports many cryptographic algorithms. The set of cryptographic algorithms supported by SB FIPS Module is given in Table 5.

Table 5: Supported Algorithms and Standards

	Algorithm	FIPS Approval	Cert #
Block Ciphers	3-Key Triple-DES (ECB, CBC, CFB64, OFB64) [SP 800-67rev2]	Approved	#2714
	2-Key Triple-DES (ECB, CBC, CFB64, OFB64) encryption [SP800-67rev2]	Disallowed	
	2-Key Triple-DES (ECB, CBC, CFB64, OFB64) decryption [SP800-67rev2]	Allowed	
	AES (ECB, CBC, CFB128, OFB128, CTR) [FIPS 197]	Approved	#5387
	DES (ECB, CBC, CFB64, OFB64)	Disallowed	
	DESX (ECB, CBC, CFB64, OFB64)	Disallowed	
	AES EAX [ANSI C12.22]	Disallowed	
	ARC2 (ECB, CBC, CFB64, OFB64) [RFC 2268]	Disallowed	
Block Cipher Modes	CMAC [NIST SP 800-38B]	Approved	#5387
	CCM [NIST SP 800-38C]	Approved	#5387
	GCM [NIST SP 800-38D]	Approved	#5387
	XTS-AES [NIST SP 800-38E]	Approved	#5387
	KW [NIST SP 800-38F]	Approved	#5387
	AES (CCM*) [ZigBee 1.0.x]	Disallowed	
Stream Cipher	ARC4	Disallowed	
Hash Functions	SHA-1 [FIPS 180-4]	Approved	#4321
	SHA-224 [FIPS 180-4]	Approved	#4321
	SHA-256 [FIPS 180-4]	Approved	#4321
	SHA-384 [FIPS 180-4]	Approved	#4321
	SHA-512 [FIPS 180-4]	Approved	#4321
	SHA-512/224 [FIPS 180-4]	Approved	#4321
	SHA-512/256 [FIPS 180-4]	Approved	#4321
	SHA3-224 [FIPS 202]	Approved	#43
	SHA3-256 [FIPS 202]	Approved	#43
	SHA3-384 [FIPS 202]	Approved	#43
	SHA3-512 [FIPS 202]	Approved	#43
	MD5 [RFC 1321]	Disallowed	
	MD4 [RFC 1320]	Disallowed	
	MD2 [RFC 1115]	Disallowed	
	AES MMO [ZigBee 1.0.x]	Disallowed	
Extended-Output Functions	SHAKE128 [FIPS 202]	Approved	#43
	SHAKE256 [FIPS 202]	Approved	#43
Message Authentication	HMAC-SHA-1 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-224 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-256 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-384 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-512 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-512/224 [FIPS 198-1]	Approved	#3568
	HMAC-SHA-512/256 [FIPS 198-1]	Approved	#3568

	HMAC-SHA3-224 [FIPS 198-1]	Approved	#3568
	HMAC-SHA3-256 [FIPS 198-1]	Approved	#3568
	HMAC-SHA3-384 [FIPS 198-1]	Approved	#3568
	HMAC-SHA3-512 [FIPS 198-1]	Approved	#3568
	HMAC-MD5 [RFC 2104]	Disallowed	
	AES-XCBC-MAC [RFC 3566]	Disallowed	
Random Number Generators	DRBG [NIST SP 800-90A]	Approved	#2085
	ANSI X9.62 RNG [ANSI X9.62]	Disallowed	
	ANSI X9.31 RNG [ANSI X9.31]	Disallowed	
	NDRNG	Allowed	
Digital Signature	DSA [FIPS 186-4] Security strength 112-256	Approved	#1391
	DSA [FIPS 186-4] Security strength 56-111	Disallowed	
	ECDSA [FIPS 186-4, ANSI X9.62] Security strength 112-256	Approved	#1423
	ECDSA [FIPS 186-4, ANSI X9.62] Security strength 80-111	Disallowed	
	RSA PKCS1 v1.5 [FIPS 186-4, PKCS #1 v2.1] Security strength 112-256	Approved	#2881
	RSA PSS [FIPS 186-4, PKCS #1 v2.1]	Approved	#2881
	RSA PKCS1 v1.5 [FIPS 186-4, PKCS #1 v2.1] Security strength 56-111	Disallowed	
	ECNR [IEEE 1363]	Disallowed	
	ECQV	Disallowed	
Key Management	DH [NIST SP 800-56A] Security strength 112-256 bits	Approved	#178
	DH [NIST SP 800-56A] Security strength 56-111 bits	Disallowed	
	ECDH [NIST SP 800-56A] Security strength 112-256	Approved	#178
	ECDH [NIST SP 800-56A] Security strength 80-111	Disallowed	
	ECMQV [NIST SP 800-56A]] Security strength 112-256	Approved	#178
	ECMQV [NIST SP 800-56A]] Security strength 80-111	Disallowed	
	ECPVS [ANSI X9.92]	Disallowed	
	ECSPEKE [IEEE 1363.2]	Disallowed	
	RSA KEM [ANSI X9.44]	Disallowed	
	RSA OAEP [NIST SP 800-56B] Security strength 112-256 bits (used in key wrapping)	Vendor Affirmed	
	RSA OAEP [NIST SP 800-56B] Security strength 56-111 bits (used in key wrapping)	Disallowed	
Components	CVL-ECDH Primitive [NIST SP 800-56A] (used in key agreement)	Approved	#1849
	CVL-RSADP [NIST SP 800-56B] (used in key wrapping)	Approved	#1851
	CVL-ANSI X9.63-2001 KDF [NIST SP 800-135] (used in ECDH key agreement)	Approved	#1850
	ECIES [ANSI X9.63]	Disallowed	

Note:
CKG (Cryptographic Key Generation, SP 800-133) is using output from DRBG for asymmetric and symmetric key generation, no post-processing.

For DSA signature generation, only the primes p and q pairs with bit lengths (2048, 224), (2048, 256) and (3072, 256) can be used in the approved mode for signature generation.

Digital signature generation using SHA-1 as its underlying hash function is disallowed.

HMAC-SHA-1 shall have a key size of at least 112 bits.

Table 6 summarizes the keys and CSPs used in the FIPS mode.

Table 6: Key and CSP, Key Size, Security Strength, and Access

Algorithm	Keys and CSPs	Key Size	Strength	Access
AES	key	128-256 bits	128-256 bits	Create, Read, Use
AES-GCM	IV	>= 96 bits	>=96 bits	Create, Use
Triple-DES	key	168 bits	112 bits	Create, Read, Use
HMAC	key	160-512 bits	128-256 bits	Use
DRBG Seed	seed	112-256 bits	112-256 bits	Use
DRBG(CTR_AES)	V and AES Key, entropy input	112-256 bits	112-256 bits	Use
DRBG(HASH)	V and C, entropy input	112-256 bits	112-256 bits	Use
DRBG(HMAC)	V and Key, entropy input	112-256 bits	112-256 bits	Use
DSA	key pair	L=2048-15360, N=224-512 bits	112-256 bits	Create, Read, Use
ECDSA	key pair	f=224-521 bits	112-256 bits	Create, Read, Use
RSA	key pair	k=2048-15360 bits	112-256 bits	Create, Read, Use
RSADP	key pair	k=2048-15360 bits	112-256 bits	Create, Read, Use
DH	static/ephemeral key pair	L=2048-15360, N=224-512 bits	112-256 bits	Create, Read, Use
ECDH	static/ephemeral key pair	f=224-521 bits	112-256 bits	Create, Read, Use
ECMQV	static/ephemeral key pair	f=224-521 bits	112-256 bits	Create, Read, Use
RSA Key wrapping	key pair	k=2048-15360 bits	112-256 bits	Create, Read, Use

4.2 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
- Loaded
- 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 state.

2. When the module is loaded on the memory, turning to the Loaded state. Then, it transitions 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 state; on failure, the module enters the Error state 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 transition to the Crypto Officer/User state when an API function is called.

4. When the API function has completed successfully, the state transitions back to Idle.

5. If the Conditional Test (Continuous RNG Test or Pair-wise Consistency Test) fails, the state transitions 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.

6 Physical Security

Physical security is not applicable to this software module at Level 1 Security.

7 Operational Environment

This module is designed for commercially available general-purpose computer operating systems such as Android or QNX. These operating systems provide modifiable environment.

This module is to be run in single user operational environment, where each user application runs in virtually separated independent space. Note that modern Operating Systems such as Android or QNX provide such operational environment, that no other process can access private and secret keys, intermediate key generation values, and other CSPs, while the cryptographic process is in use.

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 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 5).

Table 7 summarizes the management of the keys and CSPs used in the FIPS mode.

Table 7: Cryptographic Key Management

Algorithm	Keys and CSPs	Generation/Input	Output	Storage	Zeroization
AES	key	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
AES-GCM	IV	IV is generated according to scenario 2 in IG A.5	API parameter	Volatile Memory	API call Power cycle
Triple-DES	key	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
HMAC	key	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
DRBG seed	Seed	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
DRBG(CTR)	V and Key, entropy input	Entropy: generated by NDRNG. V and Key: internally generated.	API parameter	Volatile Memory	API call Power cycle
DRBG(HASH)	V and C, entropy input	Entropy: generated by NDRNG. V and C: internally generated.	API parameter	Volatile Memory	API call Power cycle
DRBG(HMAC)	V and Key, entropy input	Entropy: generated by NDRNG. V and Key: generated.	API parameter	Volatile Memory	API call Power cycle
DSA	key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
ECDSA	key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
RSA	key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
RSADP	key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
DH	static/ephemeral key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
ECDH	static/ephemeral key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
ECMQV	static/ephemeral key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle
RSA Key wrapping	key pair	Internally Generated or Input	API parameter	Volatile Memory	API call Power cycle

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

To comply with FIPS 140-2, AES-GCM must generate IV internally with Approved DRBG. DRBG seed also must be generated internally. The minimum 96-bit IV length is enforced by the module.

8.1 Key Generation

SB FIPS Module provides FIPS 140-2 compliant key generation. The underlying random number generation uses a FIPS Approved method, a DRBG (Hash, HMAC or Counter).

The module also supports ANSI X9.62 and ANSI X9.31 RNGs, however, the use of ANSI X9.62/ANSI X9.31 RNG is non-approved for key generation. No keys generated using ANSI X9.62/ANSI X9.31 RNG can be used to protect sensitive data in the Approved mode.

8.2 Key Establishment

SB FIPS Module provides the following FIPS Approved or allowed key establishment techniques [5]:

1. Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
2. EC Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
3. ECMQV (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
4. RSA (key wrapping; key establishment methodology provides between 112 and 256 bits of encryption strength)
5. KTS (AES Cert. #5387; key establishment methodology provides between 128 and 256 bits of encryption strength)

It is the 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 is a low-level cryptographic toolkit, and as such does not provide key storage.

8.5 Zeroization of Keys and CSPs

SB FIPS Module provides zeroizable interfaces which implement zeroization functions (see Table 4). Zeroization of keys and CSPs must be performed by calling the corresponding destroy functions of the objects when no longer needed.

Internally, SB FIPS Module zeroize all intermediate keys and CSPs upon each function returns.

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 without any intervention from an application. It does not involve any inputs by the operator. The following tests are applied:

1. **Software Integrity Test:**
   The software integrity test deploys HMAC SHA-256 to verify the integrity of the module.

2. **Known Answer Tests (KATs):**
   The following KATs are performed:

   AES-ECB Encrypt KAT,
   AES-ECB Decrypt KAT,
   AES-GCM Encrypt KAT,
   AES-GCM Decrypt KAT,
   AES-KW Wrapping KAT,
   AES-KW Unwrapping KAT,
   DRBG-CTR KAT,
   DRBG-HASH KAT,
   DRBG-HMAC KAT,
   DH KAT,
   DSA PCT,
   ECDH KAT,
   ECDSA PCT,
   HMAC SHA-1 KAT,
   HMAC SHA-224 KAT,
   HMAC SHA-256 KAT,
   HMAC SHA-384 KAT,
   HMAC SHA-512 KAT,
   HMAC SHA-512/224 KAT,
   HMAC SHA-512/256 KAT,
   HMAC SHA3-224 KAT,
   HMAC SHA3-256 KAT,
   HMAC SHA3-384 KAT,
   HMAC SHA3-512 KAT,
   ANS X9.63-2001 KDF KAT,
   RSA Signature Generation KAT,
   RSA Signature Verification KAT,
   RSADP Encrypt KAT,
   RSADP Decrypt KAT,
   Triple-DES-ECB Encrypt KAT, and
   Triple-DES-ECB Decrypt KAT.

   Pairwise-consistency test(PCT) is used for DSA and ECDSA. ECMQV tests are covered by the underlying arithmetic tests via ECDSA PCT and ANS X9.63-2001 KDF KAT. Each SHA KAT is considered covered by the corresponding HMAC SHA KAT. SP 800-56A Primitive “Z” Computation KATs are covered by DH KAT and ECDH KAT. The SP 800-56B Self-Tests per IG D.4 are covered by RSADP Encrypt KAT and RSADP Decrypt KAT.

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

Continuous RNG Tests for NDRNG,
Continuous RNG Tests for DRBG-CTR,
Continuous RNG Tests for DRBG-HASH,
Continuous RNG Tests for DRBG-HMAC,
Pair-wise Consistency Tests for DH key generation,
Pair-wise Consistency Tests for DSA key generation,
Pair-wise Consistency Tests for ECDH key generation,
Pair-wise Consistency Tests for ECDSA key generation,
Pair-wise Consistency Tests for ECMQV key generation, and
Pair-wise Consistency Tests for RSA key generation.

9.3 Critical Function Tests

For DRBG (CTR, HASH, and HMAC), the module implements the following critical function tests:

- SP 800-90A DRBG Instantiate Health Test
- SP 800-90A DRBG Generate Health Test
- SP 800-90A DRBG Reseed Health Test
- SP 800-90A DRBG Uninstantiate Health Test

9.4 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 an error code, and goes into the Error state.

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 Subversion (SVN) 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 is 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. Timing Attack on RSA

11.1 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 technique that requires no inversion to remove (unlike other blinding methods e.g. BSAFE Crypto-C User Manual v 4.2).

Note that Remote Timing Attacks are practical:
http://crypto.stanford.edu/ dabo/papers/ssl-timing.pdf

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.

Place the object in an appropriate location on the computer hardware for your development and runtime environments.

A.1.2 Uninstalling

Remove the object from the computer hardware.

A.2 Commands

A.2.1 Load

SB FIPS Module typically is a shared library. Depending on the platform, the module can be loaded on the memory automatically when an executable is run or calling a loader system function by an executable. Once the module is loaded, a series of self-tests will be automatically started without any intervention from an application. These tests examine the integrity of the object, and the correct operation of the cryptographic algorithms. If these tests are successful, a value of *SB_SUCCESS* will be returned and the module will be enabled.

A.2.2 Unload

When the executable which loaded SB FIPS Module finishes running, the module will be unloaded from the memory.

A.2.3 Self-Tests

sbg_FIPS140RunTest()
This user callable function runs a series of self-tests, and return *SB_SUCCESS* if the tests are successful. These tests examine the integrity of the object, and the correct operation of the cryptographic algorithms. If these tests fail, the module will be disabled. Section A.3 of this document describes how to recover from the disabled state.

A.2.4 Show Status/Mode

sbg_FIPS140GetState()
This function will return the current state of the module.

A.3 When Module is Disabled

When SB FIPS Module becomes disabled, uninstall the module and re-install it. If the module is loaded successfully by this re-installation, the recovery is successful. If this recovery attempt fails, it indicates a fatal error. Please contact Certicom Support immediately.

A.4 Operation Limitation for FIPS approved mode

For 3-key Triple-DES, NIST SP 800-67rev1 requires a limit that the same Triple-DES key shall not be used for more than 2^{28} encryptions.