Mocana Cryptographic Suite B Module

Software Version 5.5fs

Security Policy

Document Version 1.3

Mocana Corporation

November 15, 2013

TABLE OF CONTENTS

1. Module Overview

The Mocana Cryptographic Suite B Module (Software Version 5.5fs) is a software only, multi-chip standalone cryptographic module that runs on a general purpose computer. The primary purpose of this module is to provide FIPS Approved cryptographic routines to consuming applications via an Application Programming Interface. The physical boundary of the module is the case of the general purpose computer. The logical boundary of the cryptographic module is the single library (libmss.a).

The cryptographic module runs on the following operating environments:

- Integrity O/S 5.0 on Freescale MPC8544ADS Development System
- iOS-5 on iPad 2
- iOS-6 on iPad 2

The cryptographic module is also supported on the following operating environments for which operational testing was not performed:

- Linux x86-64 Kernel version 2.6.32-38 (Ubuntu 10.04)

Note: the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

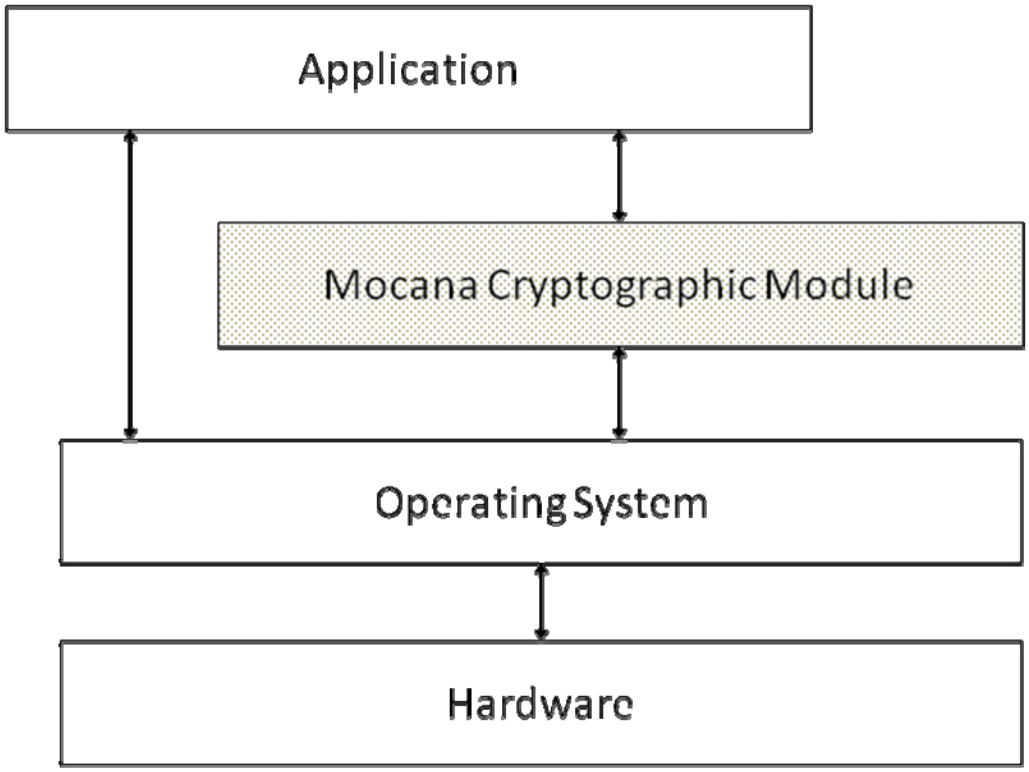

Figure 1 – Cryptographic Module Interface Diagram

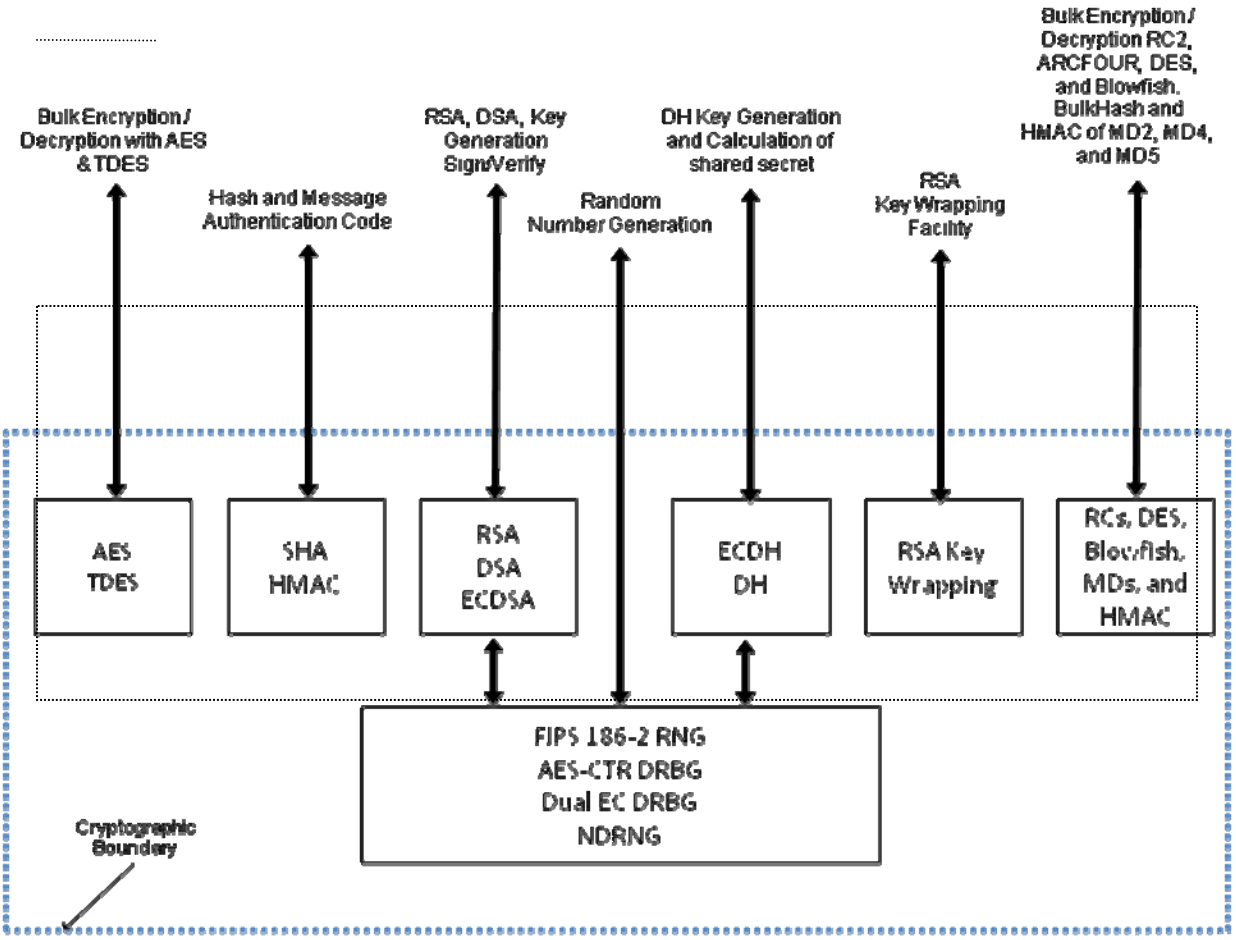

Figure 2 – Logical Cryptographic Boundary

2. Security Level

The cryptographic module meets the overall requirements applicable to Security Level 1 of FIPS 140-2.

Table 1 - Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	1
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

3. Modes of Operation

Approved mode of operation

The module supports multiple Approved modes of operation. During module initialization, a consuming application can configure the module to utilize all, or any subset of the following FIPS Approved algorithms:

- AES (ECB, CBC, OFB, CFB, CTR and GCM modes; E/D; 128, 192 and 256) – Certs. #2356 and #2096
- AES (CCM, CMAC 128, 192 and 256) – Certs. #2356 and #2096
- AES XTS (128 and 256) – Certs. #2356 and #2096
- Triple-DES (3-key and 2-key; TCBC mode; E/D) – Cert. #1333
- HMAC-SHA-1 – Cert. #1271
- HMAC-SHA-224 - Cert. #1271
- HMAC-SHA-256 - Cert. #1271
- HMAC-SHA-384 - Cert. #1271
- HMAC-SHA-512 - Cert. #1271
- SHA-1 - Cert. #1820
- SHA-224 - Cert. #1820
- SHA-256 - Cert. #1820
- SHA-384 - Cert. #1820
- SHA-512 - Cert. #1820
- RSA key generation, signature generation and verification (Gen Key X9.31; PKCS #1 1.5, Sig Gen and Sig Ver: 1024, 1536, 2048, 3072, 4096; PSS Sig Gen and Sig Ver: 1024, 1536, 2048, 3072, 4096) - Cert. #1075
- DSA key generation, signature generation and verification (PQG Gen/Ver, Key Pair Gen, Sig Gen/Ver; 1024) - Cert. #655
- ECDSA key generation, public key validation, signature generation and verification (CURVES P; 192, 224, 256, 384, 521) - Cert. #307
- FIPS 186-2 RNG - Cert. #1078
- AES-CTR based DRBG - Cert. #221
- Dual EC DRBG - Cert. #221

Non-FIPS Approved Algorithms

Within the FIPS Approved mode of operation, the module supports the following allowed algorithms:

- Diffie-Hellman (for key agreement; provides 80 or 112 bits of encryption strength)
- RSA Key Wrapping (provides between 80 and 128 bits of encryption strength)
- ECDH (for key agreement; provides between 80 and 256 bits of encryption strength)
- NDRNG

Non-FIPS Approved mode:

In addition to the above algorithms, the following algorithms are available in the non-FIPS Approved mode of operation:

- DES, Blowfish, ARC2, ARC4, MD2, MD4, MD5, HMAC-MD5, AES EAX, AES XCBC
- RSA PKCS #1 v2.1 RSAES-OAEP encryption/decryption

During operation, the module can switch service by service between an Approved mode of operation and a non-Approved mode of operation. The module will transition to the non-Approved mode of operation when one of the above non-Approved security functions is utilized in lieu of an Approved one. The module can transition back to the Approved mode of operation by utilizing an Approved security function

4. Ports and Interfaces

The physical ports of the module are provided by the general purpose computer on which the module is installed. The logical interfaces are defined as the API of the cryptographic module. The module's API supports the following logical interfaces: data input, data output, control input, and status output.

5. Identification and Authentication Policy

Assumption of roles

The Mocana Cryptographic Suite B Module shall support two distinct roles (User and Cryptographic Officer). The cryptographic module does not provide any identification or authentication methods of its own. The Cryptographic Officer and the User roles are implicitly assumed based on the service requested.

Table 2 - Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
User	N/A	N/A
Cryptographic Officer	N/A	N/A

6. Access Control Policy

Roles and Services

Table 3 – Services Authorized for Use in the Approved modes of operation

Role	Authorized Services
User	• Self-tests • Show Status • Read Version
Cryptographic-Officer	• DH Key Generation • DH Key Exchange • ECDH Key Exchange • RSA Key Generation • RSA Signature Generation • RSA Signature Verification • RSA Key Wrapping Encryption • RSA Key Wrapping Decryption • DSA Key Generation • DSA Signature Generation • DSA Signature Verification • ECDSA Key Generation • ECDSA Signature Generation • ECDSA Signature Verification • AES Encryption • AES Decryption • AES Message Authentication Code • TDES Encryption • TDES Decryption • SHA-1 • SHA-224/256 • SHA-384/512 • HMAC-SHA1 Message Authentication Code • HMAC-SHA224/256 Message Authentication Code • HMAC-SHA384/512 Message Authentication Code • FIPS 186-2 Random Number Generation • AES-CTR DRBG Random Number Generation • Dual EC DRBG Random Number Generation • Key Destruction

Other Services

Table 4 – Services Authorized for Use in the non-Approved mode of operation

Role	Authorized Services
User	• Self-tests • Show Status • Read Version
Cryptographic-Officer	• DES Encryption • DES Decryption • Blowfish Encryption • Blowfish Decryption • ARC2, ARC4 Encryption • ARC2, ARC4 Decryption • MD2 Hash • MD4 Hash • MD5 Hash • HMAC-MD5 Message Authentication Code • AES EAX Encryption • AES EAX Decryption • AES XCBC Encryption • AES XCBC Decryption • RSA PKCS #1 v2.1 RSAES-OAEP Encryption • RSA PKCS #1 v2.1 RSAES-OAEP Decryption

The cryptographic module supports the following service that does not require an operator to assume an authorized role:

- Self-tests: This service executes the suite of self-tests required by FIPS 140-2. It is invoked by reloading the library into executable memory.

Definition of Critical Security Parameters (CSPs)

The following are CSPs that may be contained in the module:

Table 5: CSP Information

Key	Description/Usage	Generation	Storage	Entry / Output	Destruction
DH Private Components	Used to derive the secret session key during DH key agreement protocol	Internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG	Temporarily in volatile RAM	N/A	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
ECDH Private Components	Used to derive the secret session key during ECDH key agreement protocol	Internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG	Temporarily in volatile RAM	N/A	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
Seed and Seed Keys	Used to seed the RNG and DRBGs for key generation	Internally via NDRNG or Externally	Temporarily in volatile RAM	Entry: Plaintext if generated externally Output: N/A	Automatically after use
RSA Private Key	Used to create RSA digital signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
RSA Key Wrapping Private Key	Used for RSA Key Wrapping decryption operation	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
DSA Private Key	Used to create DSA digital signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.

Key	Description/Usage	Generation	Storage	Entry / Output	Destruction
ECDSA Private Key	Used to create DSA digital signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Entry: Plaintext if generated externally Output: Plaintext	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
TDES Key	Used during TDES encryption and decryption	Externally.	Temporarily in volatile RAM	Entry: Plaintext Output: N/A	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
AES Keys	Used during AES encryption, decryption, and CMAC operations	Externally.	Temporarily in volatile RAM	Entry: Plaintext Output: N/A	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.
HMAC Keys	Used during HMAC-SHA-1, 224, 256, 384, 512 operations	Externally.	Temporarily in volatile RAM	Entry: Plaintext Output: N/A	An application program which uses the API may destroy the key. The Key Destruction service zeroizes this CSP.

Definition of Public Keys:

The following are the public keys contained in the module:

Table 6: Public Key Information

Key	Description/Usage	Generation	Storage	Entry/Output
DH Public Component	Used to derive the secret session key during DH key agreement protocol	Internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG	Temporarily in volatile RAM	Entry: Receive Client Public Component during DH exchange. Output: Transmit Host Public Component during DH exchange
ECDH Public Component	Used to derive the secret session key during ECDH key agreement protocol	Internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG	Temporarily in volatile RAM	Entry: Receive Client Public Component during DH exchange. Output: Transmit Host Public Component during DH exchange
RSA Public Keys	Used to verify RSA signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Input: Plaintext if generated externally t Output: Plaintext
RSA Key Wrapping Public Keys	Used for RSA Key Wrapping encryption operation	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Input: Plaintext if generated externally Output: Plaintext
DSA Public Keys	Used to verify DSA signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Input: Plaintext if generated externally Output: Plaintext

Key	Description/Usage	Generation	Storage	Entry/Output
ECDSA Public Keys	Used to verify ECDSA signatures	May be generated internally using the FIPS 186-2 RNG, AES-CTR DRBG, or Dual EC DRBG or generated externally	Temporarily in volatile RAM	Input: Plaintext if generated externally Output: Plaintext

Definition of CSPs Modes of Access

The following table defines the relationship between access to CSPs and the different module services.

Table 7 – CSP Access Rights within Roles & Services

Role		Service	Cryptographic Keys and CSPs Access Operation
C.O.	**User**		
X		DH Key Generation	Use DH Parameters Generate DH Key pair
X		DH Key Exchange	Use DH Private Component Generate DH shared secret
X		ECDH Key Exchange	Use ECDH Private Component Generate ECDH shared secret
X		RSA Key Generation	Generate RSA Public/Private Key pair
X		RSA Signature Generation	Use RSA Private Key Generate RSA Signature
X		RSA Signature Verification	Use RSA Public Key Verify RSA Signature
X		RSA Key Wrapping Encryption	Use RSA Public Key Performs Key Wrapping Encryption
X		RSA Key Wrapping Decryption	Use RSA Private Key Performs Key Wrapping Decryption
X		DSA Key Generation	Generate DSA Key Pair for Signature Generation/Verification
X		DSA Signature Generation	Use DSA Private Key Generate DSA Signature
X		DSA Signature Verification	Use DSA Public Key Verify DSA Signature
X		ECDSA Key Generation	Generate ECDSA Key Pair for Signature Generation/Verification
X		ECDSA Signature Generation	Use DSA Private Key Generate ECDSA Signature
X		ECDSA Signature Verification	Use ECDSA Public Key Verify ECDSA Signature
X		AES Encryption	Use AES Key
X		AES Decryption	Use AES Key
X		AES Message	Use AES Key

Role		Service	Cryptographic Keys and CSPs Access Operation
C.O.	**User**		
		Authentication Code	
X		TDES Encryption	Use TDES Key
X		TDES Decryption	Use TDES Key
X		SHA-1	Generate SHA-1 Output; no CSP access
X		SHA-224/256	Generate SHA-224/256 Output; no CSP access
X		SHA-384/512	Generate SHA-384/512 Output; no CSP access
X		HMAC-SHA-1 Message Authentication Code	Use HMAC-SHA-1 Key Generate HMAC-SHA-1 Output
X		HMAC-SHA-224/256 Message Authentication Code	Use HMAC-SHA-224/256 Key Generate HMAC-SHA-224/256 Output
X		HMAC-SHA-384/512 Message Authentication Code	Use HMAC-SHA-384/512 Key Generate HMAC-SHA-384/512 Output
X		FIPS 186-2 Random Number Generation	Use Seed Key to generate random number Destroy Seed Key after use
X		AES-CTR DRBG Random Number Generation	Use Seed Key to generate random number Destroy Seed Key after use
X		Dual EC DRBG Random Number Generation	Use Seed Key to generate random number Destroy Seed Key after use
X		Key Destruction	Destroy All CSPs
	X	Show Status	N/A
	X	Self-Tests	N/A
	X	Read Version	N/A

7. Operational Environment

The FIPS 140-2 Area 6 Operational Environment requirements are applicable because the Mocana Cryptographic Suite B Module operates in a modifiable operational environment.

Operational testing of the module was performed on the following environments:

- Integrity O/S 5.0
- iOS-5
- iOS-6

The module is delivered as a binary library to be linked into an application in a similar manner to the shared library version of the Mocana cryptographic module. Using this library when building the application, in contrast to a shared library, means that the integrity of the binary code in the library must be tested within the fully linked executable code of the application that incorporates it.

The module is delivered as a library file (e.g. “libmss.a”) compatible with the target execution environment of the application.

A signature file (e.g. “libmss.a.sig”) is delivered that ensures that the above library file was not changed (HMAC-SHA1 fingerprint);

An executable development tool (e.g. “secure_link”) is delivered that is compatible with the host development environment in which the application is being built;

To ensure an unbroken “chain” of integrity checks, the “secure_link” development tool must perform a self integrity check to ensure that it has not been modified. It will also verify the delivered library file and signature file. It will then act as a front end to the normal development tool linker to perform the final link with this library into an application. Finally it will sign the cryptographic-module within the executable and store the signature into the executable. If the developer does not use the “secure_link” development tool to perform the final link step of their application, the cryptographic module will fail its startup integrity check. The “secure_link” development tool performs these steps:

1.Perform integrity check on itself using the same procedure as that to be used by the developer's application as described below;

2.Locate the library file and its signature file;

3.Validate the signature with the data stored in the library file (HMAC-SHA1 finger print of whole file);

4.Execute the linking step of the application with the validated library (using the linker and linker options as specified by the developer);

5.Create a new HMAC-SHA1 fingerprint including the sections (e.g. “.text” and “.rodata”) of the created executable file that covers the library crypto code and constants. This step excludes the application’s code and constants, and any other sections by fingerprinting between top and bottom markers that are used to wrap the library code and constants;

6."Stamp" the executable file with the generated HMAC-SHA1 value, so during the application startup the integrity of the library code and constants can be validated;

Integrity Check at Application Start

During application startup, the application will check the integrity of the library code and constants during the module startup function. It verifies the integrity by executing the HMAC-SHA1 fingerprint algorithm in memory and comparing the result with the value found in a specific place in the executable application. This is analogous to the HMAC-SHA1 calculation and verification against the signature file utilized in the shared object version of the Mocana Cryptographic Suite B Module.

8. Security Rules

The Mocana Cryptographic Suite B Module design corresponds to the following security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The cryptographic module shall provide two distinct roles. These are the User role and the Cryptographic Officer role.
2. The cryptographic module does not provide any operator authentication.
3. The cryptographic module shall encrypt/decrypt message traffic using the Triple-DES or AES algorithms.
4. The cryptographic module shall perform the following self-tests:

   Power-up Self-Tests:

   - Cryptographic Algorithm Tests:
     - AES-ECB, CBC, OFB. CFB, CCM, , CTR, GCM, and XTS Encrypt and Decrypt Known Answer Tests
     - AES-CMAC Generation and Verification Known Answer Tests
     - Triple-DES CBC Encrypt and Decrypt Known Answer Tests
     - HMAC-SHA-1 Known Answer Test
     - HMAC-SHA-224 Known Answer Test
     - HMAC-SHA-256 Known Answer Test
     - HMAC-SHA-384 Known Answer Test
     - HMAC-SHA-512 Known Answer Test
     - SHA-1 Known Answer Test
     - SHA-224 Known Answer Test
     - SHA-256 Known Answer Test
     - SHA-384 Known Answer Test
     - SHA-512 Known Answer Test
     - RSA Encrypt and Decrypt Known Answer Tests
     - RSA Sign and Verify Known Answer Tests
     - DSA Pairwise Consistency Test

  - ECDSA Pairwise Consistency Test
  - ECDH Pairwise Consistency Test
  - DH Pairwise Consistency Test
  - FIPS 186-2 RNG Known Answer Test
  - AES-CTR DRBG Known Answer Test
  - Dual EC DRBG Known Answer Test
- Software Integrity Test: HMAC-SHA-1
- Critical Functions Tests: N/A

Conditional Tests:

- DSA Pairwise Consistency Test
- RSA Pairwise Consistency Test
- ECDSA Pairwise Consistency Test
- FIPS 186-2 RNG Continuous Test
- AES-CTR DRBG Continuous Test
- Dual EC DRBG Continuous Test
- NDRNG Continuous Test

The module can be configured to utilize all or only a subset of the Approved security functions listed in Section 3 above. Only the self-tests of the algorithms that are to be utilized will be run at power-up. Upon re-configuration from one Approved mode of operation to another, the module will reinitialize and perform all power-up self-tests associated with the new Approved mode of operation.

5. At any time, the operator shall be capable of commanding the module to perform the power-up self-tests by reloading the cryptographic module into memory.
6. The cryptographic module is available to perform services only after successfully completing the power-up self-tests.
7. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
8. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
9. The module shall not support concurrent operators.

10. DES, Blowfish, ARC2, ARC4, MD2, MD4, MD5, HMAC-MD5, AES EAX, AES XCBC, and RSA PKCS #1 v2.1 RSAES-OAEP encryption/decryption are not allowed for use in the FIPS Approved mode of operation. When these algorithms are used, the module is no longer operating in the FIPS Approved mode of operation. It is the responsibility of the consuming application to zeroize all keys and CSPs prior to and after utilizing these non-Approved algorithms. CSPs shall not be shared between the Approved and non-Approved modes of operation.

9. Physical Security

The FIPS 140-2 Area 5 Physical Security requirements are not applicable because the Mocana Cryptographic Suite B Module is software only.

10. Mitigation of Other Attacks Policy

The module has not been designed to mitigate any specific attacks outside the scope of FIPS 140-2 requirements.

11. Cryptographic Officer Guidance

The operating system running the Mocana Cryptographic Suite B Module must be configured in a single-user mode of operation.

Key Destruction Service

There is a context structure associated with every cryptographic algorithm available in this module. Context structures hold sensitive information such as cryptographic keys. These context structures must be destroyed via respective API calls when the application software no longer needs to use a specific algorithm any more. This API call will zeroize all sensitive information including cryptographic keys before freeing the dynamically allocated memory. See the *Mocana Cryptographic API Reference* for additional information.

12. Definitions and Acronyms

AES	Advanced Encryption Standard
API	Application Program Interface
CO	Cryptographic Officer
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator

DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman Algorithm
TDES	Triple-DES
SHA	Secure Hash Algorithm
SO	Shared Object