Riverbed Cryptographic Security Module
version 1.0

FIPS 140-2 Level 1
Non-Proprietary Security Policy

November 7, 2016

Table of Contents

Modification History

2016-11-07	Correction of OS name for platforms 37,38,43,44
2016-10-25	Addition of twenty platforms
2015-11-20	Deprecation of X9.31 RNG and Dual EC DRBG
2014-08-22	Addition of nine platforms
2014-03-18	Addition of five platforms
2014-02-20	Multiple changes to separate the Approved services from those that are non-Approved per the SP 800-131A transition
2014-01-06	Note Dual EC DRBG is non-approved in Section 4
2013-11-25	Changes to Sections 4, 5, 6
2013-08-23	Added four new platforms
2013-04-10	Initial draft

1 Introduction

This document comprises the non-proprietary FIPS 140-2 Security Policy for the Riverbed Cryptographic Security Module v1.0, hereafter referred to as the Module.

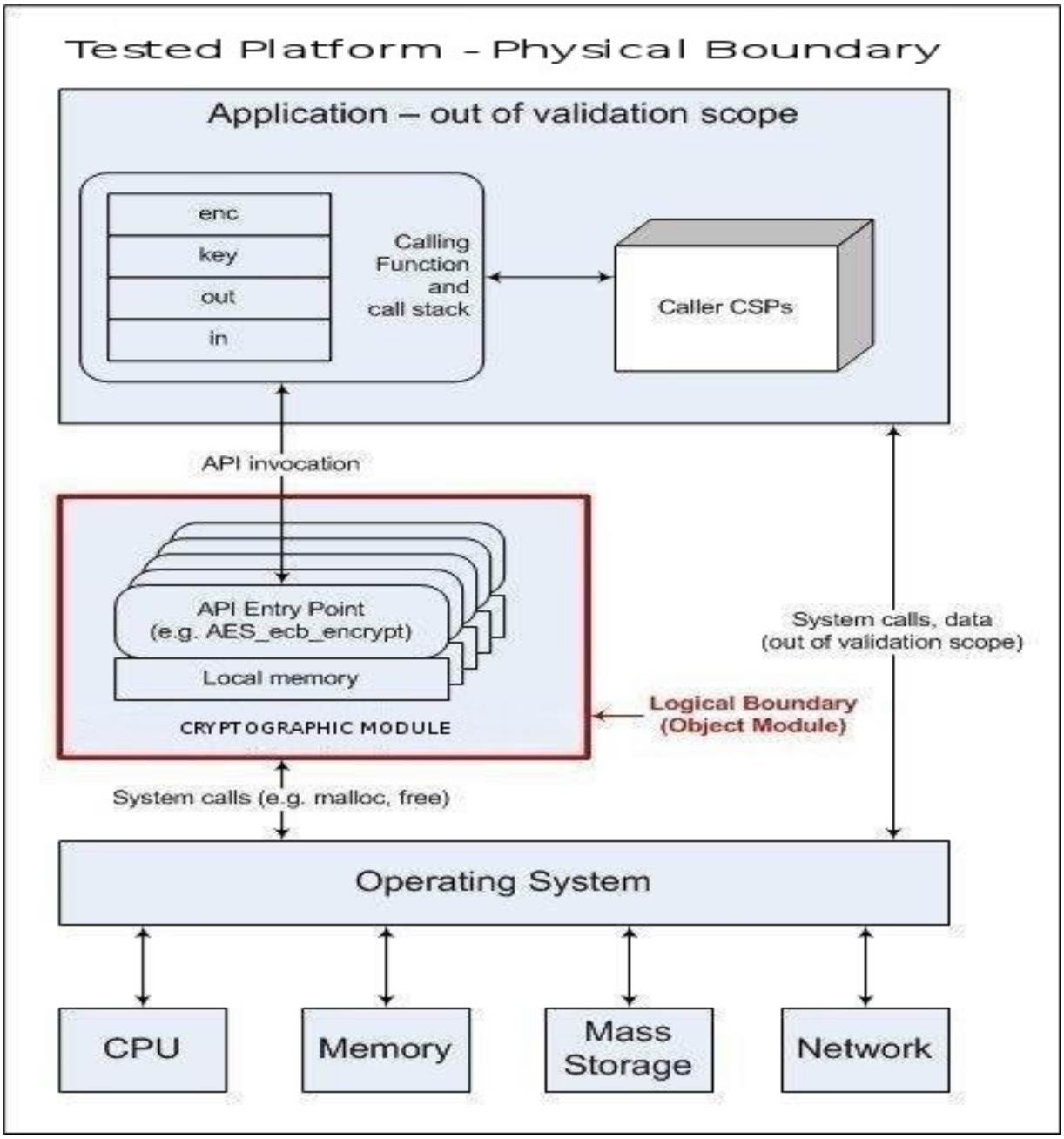
The Module is a software library providing a C-language application program interface (API) for use by other processes that require cryptographic functionality. The Module is classified by FIPS 140-2 as a software module, multi-chip standalone module embodiment. The physical cryptographic boundary is the general purpose computer on which the module is installed. The logical cryptographic boundary of the Module is the fipscanister object module, a single object module file named *fipscanister.o*. The Module performs no communications other than with the calling application (the process that invokes the Module services).

The FIPS 140-2 security levels for the Module are as follows:

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

Table 1 – Security Level of Security Requirements

The Module's software version for this validation is v1.0.



Block Diagram

2 Tested Configurations

	Operational Environment	Processor	Optimizations (Target)
1	RiOS 8.0 on Steelhead Appliance 32-bit	Intel Xeon (x86_64)	None
2	RiOS 8.0 on Steelhead Appliance 64-bit	Intel Xeon (x86_64)	None
3	RiOS 8.0 64-bit under VMware ESXi 5.1	Intel Xeon E3-1220v2 (x86_64)	None
4	RiOS 8.0 64-bit under VMware ESXi 5.1	Intel Xeon E3-1220v2 (X86_64)	AES-NI
5	Stingray OS version 4.0 64-bit under VMware ESXi 5.1	Intel Xeon E3-1220v2 (x86_64)	None
6	Stingray OS version 4.0 64-bit under VMware ESXi 5.1	Intel Xeon E3-1220v2 (x86_64)	AES-NI
7	RiOS 8.0 on Steelhead Appliance 64-bit	Intel Xeon E31220 (x86_64)	AES-NI
8	Granite OS 2.0 on Riverbed GCA-02000	AMD Opteron 4122 (x86_64)	None
9	Granite OS 2.0 under VMware ESXi 5.1	Intel Xeon E31220 (x86_64)	None
10	Granite OS 2.0 under VMware ESXi 5.1	Intel Xeon E31220 (x86_64)	AES-NI
11	Whitewater OS 3.0	Intel Xeon E5620 (x86_64)	None
12	Whitewater OS 3.0	Intel Xeon E5620 (x86_64)	AES-NI
13	Whitewater OS 3.0 under VMware ESXi 5.1	Intel Xeon E31220 (x86_64)	None
14	Whitewater OS 3.0 under VMware ESXi 5.1	Intel Xeon E31220 (x86_64)	AES-NI
15	Interceptor OS 4.5	AMD Opteron 2376 (x86_64)	None
16	RiOS 8.6 32-bit	Intel Xeon E31220 (x86_64)	None
17	RiOS 8.6 64-bit	Intel Xeon E31220(x86_64)	None
18	RiOS 8.6 64-bit under VMware ESXi 5.1	Intel Xeon E5-2430L (x86_64)	None
19	RiOS 8.6 64-bit under VMware ESXi 5.1	Intel Xeon E5-2430L (X86_64)	AES-NI

20	RiOS 8.6 64-bit	Intel Xeon E31220(x86_64)	AES-NI
21	Steelhead Mobile Controller 4.6	Intel Xeon (x86_64)	None
22	Steelhead Mobile Controller 4.6	Intel Xeon (x86_64)	AES-NI
23	Steelhead Mobile Controller 4.6 under VMware ESXi 5.1	Intel Xeon E5-2430L (x86_64)	None
24	Steelhead Mobile Controller 4.6 under VMware ESXi 5.1	Intel Xeon E5-2430L (x86_64)	AES-NI
25	RiOS 9.2 x86 64-bit	Intel Xeon E3 (x86)	AES-NI
26	RiOS 9.2 x86 64-bit	Intel Xeon E3 (x86)	None
27	RiOS 9.2 x86 under VMware ESXi 5.5	Intel Xeon E5 (x86)	AES-NI
28	RiOS 9.2 x86 under VMware ESXi 5.5	Intel Xeon E5 (x86)	None
29	RiOS 9.2 x86 64bit under KVM 1.0	Intel Xeon E5 (x86)	AES-NI
30	RiOS 9.2 x86 64bit under KVM 1.0	Intel Xeon E5 (x86)	None
31	SteelCentral Controller for SteelHead Mobile 5.0 under VMware ESXi 5.5	Intel Xeon E5 (x86)	AES-NI
32	SteelCentral Controller for SteelHead Mobile 5.0 under VMware ESXi 5.5	Intel Xeon E5 (x86)	None
33	SteelFusion 4.3 under VMware ESXi 5.5	Intel Xeon E5 (x86)	AES-NI
34	SteelFusion 4.3 under VMware ESXi 5.5	Intel Xeon E5 (x86)	None
35	Riverbed License Manager 1.0 under VMware ESXi 5.5	Intel Xeon E5 (x86)	AES-NI
36	Riverbed License Manager 1.0 under VMware ESXi 5.5	Intel Xeon E5 (x86)	None
37	Riverbed SteelCentral AppResponse 11.2 64-bit under VMware ESXi 5.5	Intel Xeon E5 (x86)	AES-NI
38	Riverbed SteelCentral AppResponse 11.2 64-bit under VMware ESXi 5.5	Intel Xeon E5 (x86)	None
39	SteelCentral Controller for SteelHead Mobile 5.0	Intel Xeon (x86)	AES-NI
40	SteelCentral Controller for SteelHead Mobile 5.0	Intel Xeon (x86)	None
41	SteelFusion 4.3	AMD Opteron 4100 Series (x86)	AES encryption acceleration
42	SteelFusion 4.3	AMD Opteron 4100 Series (x86)	None
43	Riverbed SteelCentral AppResponse 11.2	Intel Xeon E5 (x86)	AES-NI
44	Riverbed SteelCentral AppResponse 11.2	Intel Xeon E5 (x86)	None

Table 2 - Supported Platforms

3 Ports and Interfaces

The physical ports of the Module are the same as the computer system on which it is executing. The logical interface is a C-language application program interface (API).

Logical interface type	Description
Control input	*API entry point and corresponding stack parameters*
Data input	*API entry point data input stack parameters*
Status output	*API entry point return values and status stack parameters*
Data output	*API entry point data output stack parameters*

Table 3 - Logical interfaces

As a software module, control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in an error state, all output on the logical data output interface is inhibited. The module is single-threaded and in error scenarios returns only an error value (no data output is returned).

4 Modes of Operation and Cryptographic Functionality

The Module supports only a FIPS 140-2 Approved mode. Tables 4a and 4b list the Approved and Non-approved but Allowed algorithms, respectively.

Function	Algorithm	Options	Cert #
Random Number Generation; Symmetric key generation	[SP 800-90] DRBG[1] Prediction resistance supported for all variations	Hash DRBG HMAC DRBG, no reseed CTR DRBG (AES), no derivation function	310
Encryption, Decryption and CMAC	[SP 800-67]	3-Key Triple-DES TECB, TCBC, TCFB, TOFB; CMAC generate and verify	1485
	[FIPS 197] AES	128/ 192/256 ECB, CBC, OFB, CFB 1, CFB 8, CFB 128, CTR, XTS; CCM; GCM; CMAC generate and verify	2374
	[SP 800-38B] CMAC [SP 800-38C] CCM [SP 800-38D] GCM [SP 800-38E] XTS		
Message Digests	[FIPS 180-3]	SHA-1, SHA-2 (224, 256, 384, 512)	2046
Keyed Hash	[FIPS 198] HMAC	SHA-1, SHA-2 (224, 256, 384, 512)	1476
Digital Signature and Asymmetric Key Generation	[FIPS 186-2] RSA	SigVer9.31, SigVerPKCS1.5, SigVerPSS (1024/1536/2048/3072/4096 with all SHA sizes) GenKey9.31, SigGen9.31, SigGenPKCS1.5, SigGenPSS (2048/3072/4096 with all SHA-2 sizes)	1229
	[FIPS 186-2] DSA	PQG Ver, Sig Ver (1024 with SHA-1 only)	745
	[FIPS 186-3] DSA	PQG Ver, Sig Ver (1024/2048/3072 with all SHA sizes) PQG Gen, Key Pair Gen, Sig Gen (2048/3072 with all SHA-2 sizes)	745
	[FIPS 186-2] ECDSA	PKG: CURVES(P-224 P-384 P-521 K-233 K-283 K-409 K-571 B-233 B-283 B-409 B-571) PKV: CURVES(P-192 P-224 P-256 P-384 P-521 K-163 K-233 K-283 K-409 K-571 B-163 B-233 B-283 B-409 B-571)	392

1 For all DRBGs the "supported security strengths" is just the highest supported security strength per [SP800-90] and [SP800-57].

	[FIPS 186-4] ECDSA	PKG: CURVES(P-224 P-256 P-384 P-521 K-224 K-256 K-384 K-521 B-224 B-256 B-384 B-521 ExtraRandomBits TestingCandidates) PKV: CURVES(ALL-P ALL-K ALL-B) SigGen: CURVES(P-224: (SHA-224, 256, 384, 512) P-256: (SHA-224, 256, 384, 512) P-384: (SHA-224, 256, 384, 512) P-521: (SHA-224, 256, 384, 512) K-233: (SHA-224, 256, 384, 512) K-283: (SHA-224, 256, 384, 512) K-409: (SHA-224, 256, 384, 512) K-571: (SHA-224, 256, 384, 512) B-233: (SHA-224, 256, 384, 512) B-283: (SHA-224, 256, 384, 512) B-409: (SHA-224, 256, 384, 512) B-571: (SHA-224, 256, 384, 512)) SigVer: CURVES(P-192: (SHA-1, 224, 256, 384, 512) P-224: (SHA-1, 224, 256, 384, 512) P-256: (SHA-1, 224, 256, 384, 512) P-384: (SHA-1, 224, 256, 384, 512) P-521: (SHA-1, 224, 256, 384, 512) K-163: (SHA-1, 224, 256, 384, 512) K-233: (SHA-1, 224, 256, 384, 512) K-283: (SHA-1, 224, 256, 384, 512) K-409: (SHA-1, 224, 256, 384, 512) K-571: (SHA-1, 224, 256, 384, 512 B-163: (SHA-1, 224, 256, 384, 512) B-233: (SHA-1, 224, 256, 384, 512) B-283: (SHA-1, 224, 256, 384, 512) B-409: (SHA-1, 224, 256, 384, 512) B-571: (SHA-1, 224, 256, 384, 512))	392
ECC CDH (CVL)	[SP 800-56A] (§5.7.1.2)	All NIST Recommended B, K and P curves except sizes 163 and 192	65

Table 4a – FIPS Approved Cryptographic Functions

The Module supports only NIST recommended curves for use with ECDSA and ECC CDH. Refer to the transition tables that will be available at the CMVP Web site (http://csrc.nist.gov/groups/STM/cmvp/.

Category	Algorithm	Description
Key Agreement	EC DH	Non-compliant (untested) DH scheme using elliptic curve, supporting all NIST Recommended B, K and P curves. Key agreement is a service provided for calling process use, but is not used to establish keys into the Module.
Key Wrapping, Unwrapping	RSA	The RSA algorithm may be used by the calling application for wrapping or unwrapping of keys. No claim is made for SP 800-56B compliance, and no CSPs are established into or exported out of the module using these services.

Table 4b – Non-FIPS Approved But Allowed Cryptographic Functions

The Module implements the following services which are Non-Approved per the SP 800-131A transition:

Function	Algorithm	Options	Cert #
Random Number Generation; Symmetric key generation	[ANSI X9.31] RNG	AES 128/192/256	1179
Random Number Generation; Symmetric key generation	[SP 80090] DRBG	Dual EC DRBG (note the Dual EC DRBG algorithm shall not be used in the FIPS Approved mode of operation)	310
Digital Signature and Asymmetric Key Generation	[FIPS 186-2] RSA	GenKey9.31, SigGen9.31, SigGenPKCS1.5, SigGenPSS (1024/1536 with all SHA sizes, 2048/3072/4096 with SHA-1)	1229
	[FIPS 186-2] DSA	PQG Gen, Key Pair Gen, Sig Gen (1024 with all SHA sizes, 2048/3072 with SHA-1)	745
	[FIPS 186-3] DSA	PQG Gen, Key Pair Gen, Sig Gen (1024 with all SHA sizes, 2048/3072 with SHA-1)	745
	[FIPS 186-2] ECDSA	PKG: CURVES(P-192 K-163 B-163) SIG(gen): CURVES(P-192 P-224 P-256 P-384 P-521 K-163 K-233 K-283 K-409 K-571 B-163 B-233 B-283 B-409 B-571)	392
	[FIPS 186-4] ECDSA	PKG: CURVES(P-192 K-163 B-163) SigGen: CURVES(P-192: (SHA-1, 224, 256, 384, 512) P-224:(SHA-1) P-256:(SHA-1) P-384: (SHA-1) P-521:(SHA-1) K-163: (SHA-1, 224, 256, 384, 512) K-233:(SHA-1) K-283:(SHA-1) K-409:(SHA-1) K-571:(SHA-1) B-163: (SHA-1, 224, 256, 384, 512) B-233:(SHA-1) B-283: (SHA-1) B-409:(SHA-1) B-571:(SHA-1))	392
ECC CDH (CVL)	[SP 800-56A] (§5.7.1.2)	All NIST Recommended B, K and P curves sizes 163 and 192	65

Table 4c – FIPS Non-Approved Cryptographic Functions

X9.31 RNG is NonApproved effective December 31, 2015, per the CMVP Notice "X9.31 RNG transition, December 31, 2015". These Non_Approved algorithms shall not be used when operating in the FIPS Approved mode of operation.

The Module supports only a FIPS 140-2 Approved mode. The Module requires an initialization sequence (see IG 9.5): the calling application invokes `FIPS_mode_set()`[2], which returns a “1” for success and “0” for failure. If `FIPS_mode_set()` fails then all cryptographic services fail from then on. The application can test to see if FIPS mode has been successfully performed.

2 The function call in the Module is `FIPS_module_mode_set()` which is typically used by an application via the `FIPS_mode_set()` wrapper function.

The Module is a cryptographic engine library, which can be used only in conjunction with additional software. Aside from the use of the NIST Recommended elliptic curves as trusted third party domain parameters, all other FIPS 186-3 assurances are outside the scope of the Module, and are the responsibility of the calling process.

4.1 Critical Security Parameters and Public Keys

All CSPs used by the Module are described in this section. All access to these CSPs by Module services are described in Section 4. The CSP names are generic, corresponding to API parameter data structures.

CSP Name	Description
RSA SGK	RSA (2048 to 16384 bits) signature generation key
RSA KDK	RSA (2048 to 16384 bits) key wrapping/unwrapping
DSA SGK	[FIPS 186-3] DSA (2048/3072) signature generation key
ECDSA SGK	ECDSA (All NIST Recommended B, K, and P curves except 163, 192 curve sizes) signature generation key
EC DH Private	EC DH (All NIST Recommended B, K, and P curves except 163, 192 curve sizes) private key agreement key.
AES EDK	AES (128/192/256) encrypt / decrypt key
AES CMAC	AES (128/192/256) CMAC generate / verify key
AES GCM	AES (128/192/256) encrypt / decrypt / generate / verify key and IV
AES XTS	AES (256/512) XTS encrypt / decrypt key
Triple-DES EDK	Triple-DES (3-Key) encrypt / decrypt key
Triple-DES CMAC	Triple-DES (3-Key) CMAC generate / verify key
HMAC Key	Keyed hash key (160/224/256/384/512)
Hash_DRBG CSPs	V (440/880 bits) and C (440/880 bits), entropy input (length dependent on security strength)
HMAC_DRBG CSPs	V (160/224/256/384/512 bits) and Key (160/224/256/384/512 bits), entropy input (length dependent on security strength)
CTR_DRBG CSPs	V (128 bits) and Key (AES 128/192/256), entropy input (length dependent on security strength)
CO-AD-Digest	Pre-calculated HMAC-SHA-1[3] digest used for Crypto Officer role authentication
User-AD-Digest	Pre-calculated HMAC-SHA-1[5] digest used for User role authentication

Table 4.1a – Critical Security Parameters

The module does not output intermediate key generation values. The module generates cryptographic keys whose strengths are modified by available entropy.

CSP Name	Description
RSA SVK	RSA (1024 to 16384 bits) signature verification public key

3 For HMAC-SHA-1 the required key size must be least 112 bits

RSA KEK	RSA (2048 to 16384 bits without SHA-1) key wrapping/unwrapping
DSA SVK	[FIPS 186-3] DSA (1024/2048/3072) signature verification key
ECDSA SVK	ECDSA (All NIST Recommended B, K and P curves except sizes 163 and 192) signature verification key
EC DH Public	EC DH (All NIST Recommended B, K and P curves except sizes 163 and 192) public key agreement key.

Table 4.1b – Public Keys

For all CSPs and Public Keys:

Storage: RAM, associated to entities by memory location. The Module stores RNG and DRBG state values for the lifetime of the RNG or DRBG instance. The module uses CSPs passed in by the calling application on the stack. The Module does not store any CSP persistently (beyond the lifetime of an API call), with the exception of RNG and DRBG state values used for the Modules' default key generation service.

Generation: The Module implements SP 800-90 compliant DRBG services for creation of symmetric keys, and for generation of DSA, elliptic curve, and RSA keys as shown in Table 4a. The calling application is responsible for storage of generated keys returned by the module. The AES GCM IV is generated using the SP 800-90A Hash DRBG, per NIST 800-38D section 8.2.2 which requires the minimum IV length of 96 bits.

Entry: All CSPs enter the Module's logical boundary in plaintext as API parameters, associated by memory location. However, none cross the physical boundary.
Output: The Module does not output CSPs, other than as explicit results of key generation services. However, none cross the physical boundary.

Destruction: Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module.

Private and secret keys as well as seeds, seed keys, and entropy input are provided to the Module by the calling application, and are destroyed when released by the appropriate API function calls. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently. An authorized application as user (Crypto-Officer and User) has access to all key data generated during the operation of the Module.

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

Module users (the calling applications) shall use entropy sources that meet the security strength required for the random number generation mechanism as shown in [SP 800-90] Table 2

(Hash_DRBG, HMAC_DRBG) and Table 3 (CTR_DRBG). This entropy is supplied by means of callback functions. Those functions must return an error if the minimum entropy strength cannot be met. The Module provides no assurance of the minimum strength of generated keys.

5 Roles, Authentication and Services

The Module meets all FIPS 140-2 level 1 requirements for Roles and Services, implementing both Crypto-User and Crypto-Officer roles. As allowed by FIPS 140-2, the Module does not support user authentication for those roles. Only one role may be active at a time and the Module does not allow concurrent operators.

The User and Crypto Officer roles are implicitly assumed by the entity accessing services implemented by the Module. The Crypto Officer can install and initialize the Module. The Crypto Officer role is implicitly entered when installing the Module or performing system administration functions on the host operating system.

- User Role (User): Loading the Module and calling any of the API functions. This role has access to all of the services provided by the Module.
- Crypto Officer Role (CO): Installation of the Module. This role is assumed implicitly when the system administrator installs the Module.

All services implemented by the Module are listed below, along with a description of service CSP access. If the module is not initialized as per Section 4 of the Security Policy, non-conformant versions of the services in Table 5 are made available to the calling application.

Service	Role	Description
Initialize	User, CO	Module initialization. Does not access CSPs.
Self-test	User, CO	Perform self tests (FIPS_selftest). Does not access CSPs.
Show status	User, CO	Functions that provide module status information: • Version (as unsigned long or const char *) • FIPS Mode (Boolean) Does not access CSPs.
Zeroize	User, CO	Functions that destroy CSPs: • fips_rand_prng_reset: destroys RNG CSPs. • fips_drbg_uninstantiate: for a given DRBG context, overwrites DRBG CSPs (Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs) All other services automatically overwrite CSPs stored in allocated memory. Stack cleanup is the responsibility of the calling application.
Random number generation	User, CO	Used for random number and symmetric key generation. • Seed or reseed an RNG or DRBG instance • Determine security strength of an RNG or DRBG instance • Obtain random data Uses and updates RNG CSPs, Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs.
Asymmetric	User, CO	Used to generate DSA, ECDSA and RSA keys:

Service	Role	Description
key generation		RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK There is one supported entropy strength for each mechanism and algorithm type, the maximum specified in SP800-90
Symmetric encrypt/decrypt	User, CO	Used to encrypt or decrypt data. Executes using AES EDK, Triple-DES EDK (passed in by the calling process).
Symmetric digest	User, CO	Used to generate or verify data integrity with CMAC. Executes using AES CMAC, Triple-DES, CMAC (passed in by the calling process).
Message digest	User, CO	Used to generate a SHA-1 or SHA-2 message digest. Does not access CSPs.
Keyed Hash	User, CO	Used to generate or verify data integrity with HMAC. Executes using HMAC Key (passed in by the calling process).
Key transport[4]	User, CO	Used to encrypt or decrypt a key value on behalf of the calling process (does not establish keys into the module). Executes using RSA KDK, RSA KEK (passed in by the calling process).
Key agreement	User, CO	Used to perform key agreement primitives on behalf of the calling process (does not establish keys into the module). Executes using EC DH Private, EC DH Public (passed in by the calling process).
Digital signature	User, CO	Used to generate or verify RSA, DSA or ECDSA digital signatures. Executes using RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK (passed in by the calling process).
Utility	User, CO	Miscellaneous helper functions. Does not access CSPs.

Table 5 - Services and CSP Access

Services that are not FIPS Approved using key sizes and CSPs specified in Tables 4a, 4b, 4.1a, 4.1b are not allowed for use in FIPS mode.

4 "Key transport" can refer to a) moving keys in and out of the module or b) the use of keys by an external application. The latter definition is the one that applies to this Module.

6 Self-test

The Module performs the self-tests listed below on invocation of Initialize or Self-test.

Algorithm	Type	Test Attributes
Software integrity	KAT	HMAC-SHA1
HMAC	KAT	One KAT per SHA1, SHA224, SHA256, SHA384 and SHA512 Per IG 9.1, this testing covers SHA POST requirements.
AES	KAT	Separate encrypt and decrypt, ECB mode, 128 bit key length
AES CCM	KAT	Separate encrypt and decrypt, 192 key length
AES GCM	KAT	Separate encrypt and decrypt, 256 key length
XTS-AES	KAT	128, 256 bit key sizes to support either the 256-bit key size (for XTS-AES-128) or the 512-bit key size (for XTS-AES-256)
AES CMAC	KAT	Sign and verify CBC mode, 128, 192, 256 key lengths
Triple-DES	KAT	Separate encrypt and decrypt, ECB mode, 3-Key
Triple-DES CMAC	KAT	CMAC generate and verify, CBC mode, 3-Key
RSA	KAT	Sign using 2048 bit key, SHA-256, PKCS#1
RSA	KAT	Verify using 2048 bit key, SHA-256, PKCS#1
DSA	PCT	Sign and verify using 2048 bit key, SHA-384
DRBG	KAT	CTR_DRBG: AES, 256 bit with and without derivation function HASH_DRBG: SHA256 HMAC_DRBG: SHA256
ECDSA	PCT	Keygen, sign, verify using P-224, K-233 and SHA512.
ECC CDH	KAT	Shared secret calculation per SP 800-56A §5.7.1.2, IG 9.6

Table 6a - Power On Self Tests (KAT = Known answer test; PCT = Pairwise consistency test)

The `FIPS_mode_set()`[5] function performs all power-up self-tests listed above with no operator intervention required, returning a “1” if all power-up self-tests succeed, and a “0” otherwise. If any component of the power-up self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to `FIPS_mode_set()`[5] succeeds.

The power-up self-tests may also be performed on-demand by calling `FIPS_selftest()`, which returns a “1” for success and “0” for failure. Interpretation of this return code is the responsibility of the calling application.

The Module also implements the following conditional and critical function tests:

Algorithm	Test
DRBG	Critical function test as required by [SP800-90] Section 11

5 `FIPS_mode_set()` calls Module function `FIPS_module_mode_set()`

Algorithm	Test
DRBG	FIPS 140-2 continuous test for stuck fault
DRBG	KAT for Dual_EC_DRBG
DSA	Pairwise consistency test on each generation of a key pair
ECDSA	Pairwise consistency test on each generation of a key pair
RSA	Pairwise consistency test on each generation of a key pair
ANSI X9.31 RNG	Continuous test for stuck fault
ANSI X9.31 RNG	KAT for 128, 192, 256 bit AES keys

Table 6b - Conditional and Critical Function Tests

In the event of a DRBG self-test failure the calling application must uninstantiate and re-instantiate the DRBG per the requirements of [SP 800-90]; this is not something the Module can do itself. The uninstantiation of the DRBG by the calling application zeroizes the internal state of the DRBG to ensure it is not accessible prior to the reinstatiation of the DRBG.

Pairwise consistency tests are performed for both possible modes of use, e.g. Sign/Verify and Encrypt/Decrypt.

7 Operational Environment

The tested operating systems segregate user processes into separate process spaces. Each process space is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the calling application, and implicitly satisfies the FIPS 140-2 requirement for a single user mode of operation.

The tested operating systems are listed in Table 2:

Granite OS 2.0
Interceptor OS 4.5
RiOS 8.0
RiOS 8.6
RiOS 9.2
Riverbed License Manager 1.0
SteelCentral AppResponse 11.0
SteelCentral Controller for SteelHead Mobile 5.0
SteelFusion 4.3
Steelhead Mobile Controller 4.6
Stingray OS version 4.0
Whitewater OS 3.0

8 Mitigation of other attacks

The module is not designed to mitigate against attacks which are outside of the scope of FIPS 140-2.