Nokia Secure Sockets Cryptographic Module

FIPS 140-2 Security Policy

Version No.: 2.14
Date: June 19, 2018

Prepared by:
Nokia of America Corporation (NoAC)
800 5th Avenue, Suite 3700
Seattle, WA 98104

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

Table of Contents

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

List of Figures

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

List of Tables

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

1 Introduction

This document is a non-proprietary FIPS 140-2 Security Policy for the Nokia Secure Sockets Cryptographic Module (the M-odule) with version 2.1 and 2.2. It contains a specification of the rules under which the Module must operate and describes how the Module meets the requirements as specified in Federal Information Processing Standards Publication 140-2 (FIPS PUB 140-2) for a Security Level 1, multi-chip, standalone software module.

1.1 Purpose of the Security Policy

There are three major reasons why a security policy is requested:

- It is required for FIPS 140-2 validation.
- It allows individuals and organizations to determine whether the cryptographic module, as implemented, satisfies the stated security policy.
- It describes the capabilities, protections, and access rights provided by the cryptographic module that will allow individuals and organizations to determine whether it meets their security requirements.

1.2 Target Audience

This document will be one of many that are submitted as a package for FIPS validation; it is intended for the following people:

- Developers working on the release
- The FIPS 140-2 testing lab
- Cryptographic Module Validation Program (CMVP)
- Consumers

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

2 Cryptographic Module Specification

This document is the non-proprietary security policy for the Nokia Secure Sockets Cryptographic Module, and was prepared as part of the requirements process that will ensure its conformance with Federal Information Processing Standard (FIPS) 140-2, Level 1. The following section describes the Module and how it complies with the FIPS 140-2 standard in each of the required areas.

2.1 Module Description

Table 1, below, provides an overview of the security level required for each validation section.

Security Component	Security 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
EMI/EMC	1
Self Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1: Security Levels

The Module has been tested by the laboratory on the platforms shown in Table 2: Tested Platforms.

Module/Implementation version	Processor	OS and Version	Test Platform
Nokia Secure Sockets Cryptographic Module 2.1	AMD Geode	Red Hat Enterprise Linux 6 32-bit (single-user mode)	oMG 2000
Nokia Secure Sockets Cryptographic Module 2.1	Intel x86 without AES-NI support	Vyatta 6.4 32-bit (single-user mode)	Dell PowerEdge R210
Nokia Secure Sockets Cryptographic Module 2.1	Intel x86 With AES-NI support	Vyatta 6.4 32-bit (single-user mode)	Dell PowerEdge R210
Nokia Secure Sockets Cryptographic Module 2.2	Intel i5 With AES-NI support	Windows 7 x86_64 native and Java support via JNI (single-user mode)	HP Pro Book 640 G1

Table 2: Tested Platforms

The only change in the version 2.2 as compared to the version 2.1 is the addition of support for the Windows platform.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

In addition to the configurations tested by the laboratory, vendor-affirmed testing was performed using Nokia Secure Sockets Cryptographic Module 2.1 on the following platforms:

- Dell PowerEdge R220 with Intel x86 (with and without AES-NI support) and Vyatta 6.4 32-bit.
- Cisco UCS C220 M3 with Intel Xeon E5 x86-64 & RHEL 6.6 64-bit running on VMware ESXi 5.1 Hypervisor.
- Cisco UCS C220 M3 with Intel Xeon E5 i686 & RHEL 6.6 64-bit running on VMware ESXi 5.1 Hypervisor.
- Cisco UCS C220 M3 with Intel Xeon E5 x86-64 & RHEL 6.7 64-bit running on VMware ESXi 5.1 Hypervisor.
- Cisco UCS C220 M3 with Intel Xeon E5 i686 & RHEL 6.7 64-bit running on VMware ESXi 5.1 Hypervisor
- oMG 2000 with AMD Geode and Linux kernel 3.4.86

Also, the vendor-affirmed testing was performed using Nokia Secure Sockets Cryptographic Module 2.2 on the following platforms:

- HP Pro Book 640 G1 with Intel i5 and Windows 7x86_64 native and Java support via JNI with 32 bit binary
- HP 600 G1 with Intel i3 and Windows 7x86_64 native and Java support via JNI with 32 bit binary
- HP 600 G1 with Intel i3 and Windows 7x86_64 native and Java support via JNI with 64 bit binary
- Linux 3.6 64-bit with Intel x64 with AES-NI running on Peplink Balance 2500
- Linux 3.6 32-bit with PowerPC running on Pepwave MAX HD4 MediaFast
- Linux 3.6 32-bit with MIPS running on Pepwave MAX BR1 MK2
- Linux 3.6 64-bit with Intel Core i5 with AES-NI running Peplink FusionHub on VMware ESXi 5.5.0 Hypervisor
- VyOS 1.6.0.0 with Linux kernel 4.4 on Dell PowerEdge R220 with Intel Xeon E3-1220v3
- VyOS 1.6.0.0 with Linux kernel 4.4 on Dell PowerEdge R230 with Intel Xeon E3-1220v5
- NetCloud OS 6 with Linux kernel 3.14 on ARM Cortex-A7 running on Cradlepoint IBR900 Series Routers
- NetCloud OS 6 with Linux kernel 3.14 on ARM Cortex-A7 running on Cradlepoint IBR1700 Series Routers
- NetCloud OS 6 with Linux kernel 3.14 on ARM Cortex-A7 running on Cradlepoint AER2200 Series Routers
- NetCloud OS 7 with Linux kernel 4.4.100 on ARM Cortex-A7 running on Cradlepoint IBR900 Series Routers
- NetCloud OS 7 with Linux kernel 4.4.100 on ARM Cortex-A7 running on Cradlepoint IBR1700 Series Routers
- NetCloud OS 7 with Linux kernel 4.4.100 on ARM Cortex-A7 running on Cradlepoint AER2200 Series Routers

Note: Per IG G.5, the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when the module is ported to the vendor affirmed platforms that are not listed on the validation certificate.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.2 Description of Approved Mode

The Module supports a FIPS-Approved mode and provides the following FIPS-Approved or Allowed functions:

- AES (128/192/256 ECB, CBC, OFB, CFB 1, CFB 8, CFB 128, CTR, XTS; CCM; GCM; CMAC)
- Triple-DES (3-key ECB, CBC, CFB, OFB; CMAC)
- SHS (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- HMAC (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- DRBGs (SP800-90A: HASH DRBG, HMAC DRBG, CTR DRBG)
- RSA (FIPS 186-2 2048 bits key for all services; 1024, 1536 bits key for signature verification only)
- DSA (FIPS 186-2, FIPS 186-3 2048,3072 bits key for all services; 1024 bits key for domain parameters verification and signature verification only)
- ECDSA (FIPS 186-2, FIPS 186-3 P-224, 256, 384, 521; K-233, 283, 409, 571; B-233, 283, 409, 571; for all services; P-192 K-163,B-163 for public key verification and signature verification only)
- KAS ECC Component (ECC CDH Primitive as specified in section 5.7.1.2 of SP 800-56A P-224, 256,384,521;K-233, 283,409,571;B-233, 283,409,571)
- RSA key wrapping (2048, 3072, 4096)

For the details of these functions including the algorithm certificate numbers, CSPs, accessing roles, etc., please see section 4.2 Services.

As per the SP800-131A transition on 2014-01-01, the key lengths providing less than 112 bits of security strength are listed as non-Approved functions. Also for the cryptographic security reason, the SP 800-90A Dual EC DRBG has been moved to non-Approved functions list even though the implementation is validated with CAVS cert. #304 and #305. The use of Dual EC DRBG and other algorithms with following key sizes is not recommended. Also, ANS X9.31 RNG is disallowed after 2015 and should not be used in FIPS mode.

- DRBGs (SP800-90A: Dual EC DRBG)
- RSA (FIPS 186-2,1024,1536 bits key for key generation and signature generation)
- DSA (FIPS 186-2, FIPS 186-3 1024 bits key for key generation and signature generation)
- ECDSA (FIPS 186-2, FIPS 186-3 Curves P-192 K-163,B-163 and Curves P-224, 256, 384, 521; K-233, 283, 409, 571; B-233, 283, 409, 571 with SHA-1 for key and signature generation)
- KAS ECC Component (ECC CDH Primitive as specified in section 5.7.1.2 of SP 800-56A Curves P-192, K-163,B-163)
- RSA key wrapping (1024)
- RNG (ANS X9.31)

The use of any of the above listed non-Approved functions will result the module operating in a non-FIPS-Approved mode.

PKCS #3 Diffie-Hellman primitive is implemented in the Module, but it is not allowed to be used in the FIPS-Approved mode. The calling application shall use the EC Diffie-Hellmann primitive service provided by the module.

2.3 Cryptographic Module Boundary

The logical boundary of the Module is the fipscanister object module, a single binary object module file *fipscanister.o*. It is distributed in the following packages for the target platforms:

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

- *coco-openssl-crypto-canister_2.1-4.6.1210_i386.deb* with HMAC-SHA-1 value *0e080ec50f21d9b96b83928bd3afec4b3e9a9042* for Dell PowerEdge R210 with Vyatta 6.4 Operating System
- *coco-openssl-crypto-canister-cross-geode-2.1-4.6.1210.i686.rpm* with HMAC-SHA-1 value *d8de6cb46396527eb64bb7b7dd4103036e17b1a4* for oMG 2000 with Red Hat Enterprise Linux 6 Operating System

- *win-7-x86_64_fipscanister.lib* with HMAC-SHA-1 value *4ebc0d697d8e35b303ee453bfa2f66c1b3cc8a87* for HP Pro Book 640 G1 with Windows 7x86_64 native and Java support via JNI

Figure 1 shows the logical boundary of the Module's software components.

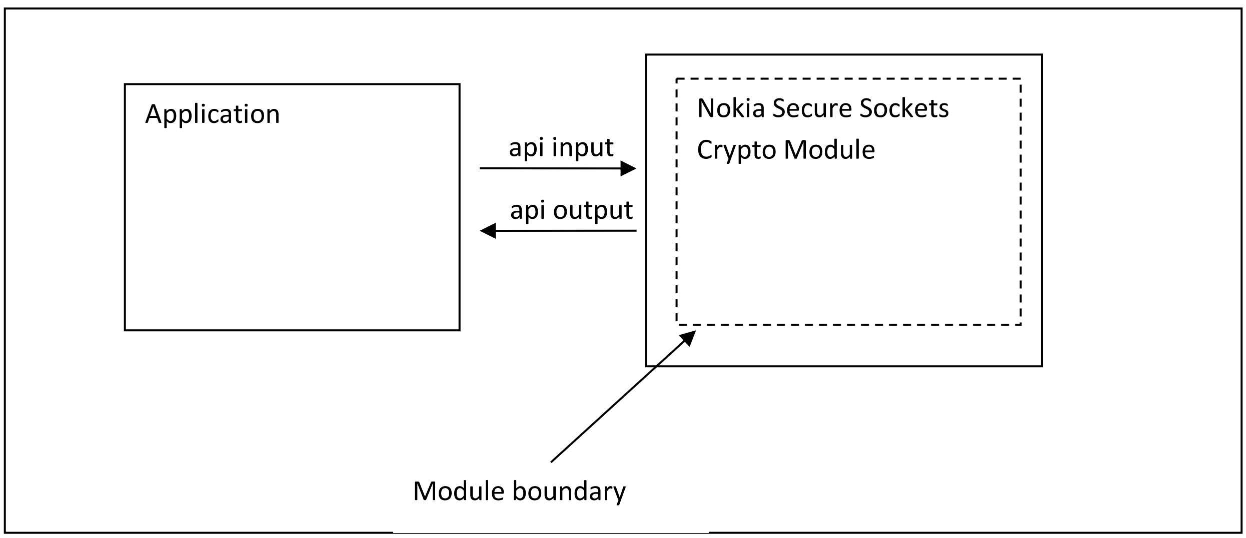

Figure 1: Software Block Diagram

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

The physical boundary of the Module is the enclosure of the general purpose computer on which the Module is installed and executed. Figure 2 shows the physical boundary of the Module and the hardware components of the test platforms on which the Module is installed and executed.

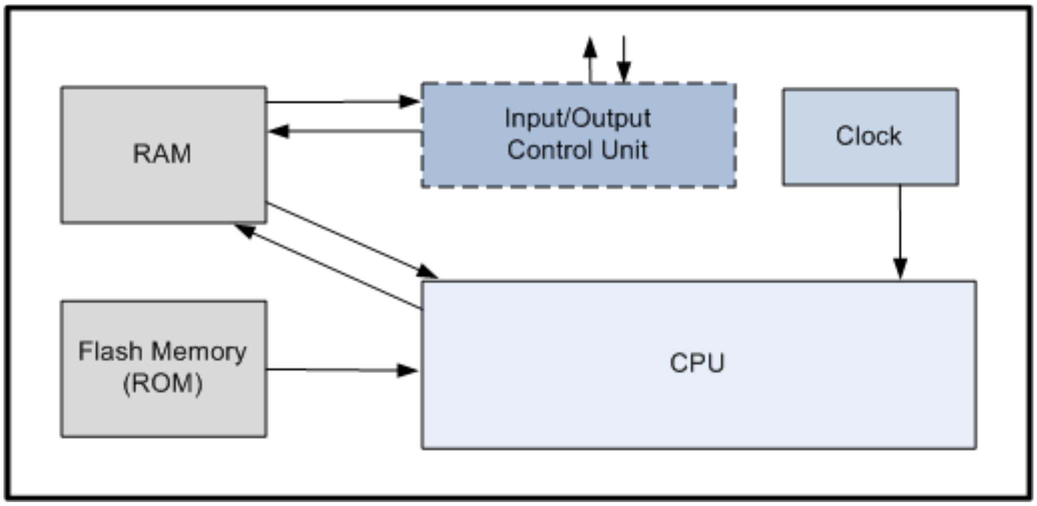

Figure 2: Hardware Block Diagram

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

3 Cryptographic Module Ports and Interfaces

The physical ports of the Module are the same as the computer system on which this software module is executing. The logical interface is an application program interface (API) as shown in Table 3, below.

FIPS Interface	Ports
Data Input	API input parameters
Data Output	API output parameters
Control Input	API function calls, HMAC-SHA-1 value in the binary code and the binary code itself on which the HMAC-SHA-1 is calculated
Status Output	API return codes and status parameters
Power Input	Physical power connector of the underlying host platform, not in the logical boundary of the Module. Included for the sake of completion.

Table 3: Ports and Interfaces

When the Module is performing self-tests or is in an error state, all output on the logical data output interface is inhibited. As a software module, it cannot control the physical ports.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

4 Roles, Services, and Authentication

4.1 Roles

The User and Crypto Officer roles are implicitly assumed by the entity that is accessing services implemented by the Module, so no further authentication is required. The services associated with each role are explained in the next section.

4.2 Services

Service	Roles		CSP	Modes	FIPS Approved (Cert #)[1]	Standard	API Functions
	User	CO					
Symmetric Algorithms							
AES encryption/ decryption	✓		128-, 192-, 256-bit keys	ECB, CBC, CFB1, CFB8, CFB128, OFB, CTR	2366 2367 2381 3474	FIPS 197	FIPS_cipherinit FIPS_cipher FIPS_cipher_ctx_new FIPS_cipher_ctx_init FIPS_cipher_ctx_ctrl FIPS_cipher_ctx_copy FIPS_cipher_ctx_set_key_length FIPS_cipher_ctx_free FIPS_cipher_ctx_cleanup
Triple-DES encryption/ decryption	✓		64-bit independent K1,K2,K3	ECB, CBC, CFB1, CFB8, CFB64, OFB	1479 1480 1959	SP 800-67	FIPS_cipherinit FIPS_cipher FIPS_cipher_ctx_new FIPS_cipher_ctx_init FIPS_cipher_ctx_ctrl FIPS_cipher_ctx_copy FIPS_cipher_ctx_set_key_length FIPS_cipher_ctx_free FIPS_cipher_ctx_cleanup
XTS encryption / decryption	✓		128-, 256-bit AES keys	Largest data length 2^16	2366 2367 2381	SP 800-38E	FIPS_cipherinit FIPS_cipher FIPS_cipher_ctx_new

[1] CAVS certificate refers to the vendor name CoCo Communications Corp., which is a prior name for Unium Inc., acquired by Nokia of America Corporation (NoAC).

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

					3474		FIPS_cipher_ctx_init FIPS_cipher_ctx_ctrl FIPS_cipher_ctx_copy FIPS_cipher_ctx_set_key_length FIPS_cipher_ctx_free FIPS_cipher_ctx_cleanup
Asymmetric Algorithms							
DSA Signature verification	✓		1024 bits modulus size key pair	With SHA-1 only	739 740 982	FIPS 186-2	FIPS_dsa_sig_new FIPS_dsa_verify FIPS_dsa_verify_ctx FIPS_dsa_verify_digest FIPS_dsa_sig_free
DSA domain parameter generation	✓		L=2048,N=224,L=2048,N=256, L=3072,N=256	With all SHA sizes except SHA-1	739 740 982	FIPS 186-4	FIPS_dsa_new FIPS_dsa_openssl fips_dsa_builtin_paramgen2 FIPS_dsa_free
DSA Key Pair generation with DRBG	✓		L=2048,N=224,L=2048,N=256,L=3072,N=256	With all SHA sizes except SHA-1	739 740 982	FIPS 186-4	FIPS_dsa_generate_key
DSA Signature generation	✓		L=2048,N=224,L=2048,N=256,L=3072,N=256	With all SHA sizes except SHA-1	739 740 982	FIPS 186-4	FIPS_dsa_sig_new FIPS_dsa_sign FIPS_dsa_sign_ctx FIPS_dsa_sign_digest FIPS_dsa_sig_free
DSA Signature verification	✓		L=1024,N=160,L=2048,N=224,L=2048,N=256, L=3072,N=256	With all SHA sizes	739 740 982	FIPS 186-4	FIPS_dsa_sig_new FIPS_dsa_verify FIPS_dsa_verify_ctx FIPS_dsa_verify_digest FIPS_dsa_sig_free
RSA key pair generation with DRBG	✓		2048, 3072, 4096 modulus size. Public key values 3, 17, and 65537.		1222 1223 1790	FIPS 186-2	FIPS_rsa_rsa FIPS_rsa_blinding_off FIPS_rsa_blinding_on FIPS_rsa_flags FIPS_rsa_pkcs1_ssleay FIPS_rsa_size FIPS_rsa_x931_derive_ex FIPS_rsa_x931_generate_key_ex FIPS_rsa_generate_key_ex FIPS_rsa_free
RSA signature Generation based on	✓		2048, 3072, 4096 bit modulus SHA-256,		1222 1223 1790	FIPS 186-2	FIPS_rsa_sign FIPS_rsa_sign_ctx FIPS_rsa_sign_digest

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

X9.31			SHA-384, SHA-512 hashing algorithm				
RSA Signature Verification based on X9.31	✓		1024, 1536, 2048, 3072, 4096 bit modulus		1222 1223 1790	FIPS 186-2	FIPS_rsa_verify FIPS_rsa_verify_ctx FIPS_rsa_verify_digest
RSA Signature Generation based on PKCS#1 v1.5	✓		2048, 3072, 4096 bit modulus	SHA-224, SHA-256, SHA-384, SHA-512	1222 1223 1790	FIPS 186-2	FIPS_rsa_sign FIPS_rsa_sign_ctx FIPS_rsa_sign_digest
RSA Signature Verification based on PKCS#1 v1.5	✓		1024, 1536, 2048, 3072, 4096 bit modulus	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1222 1223 1790	FIPS 186-2	FIPS_rsa_verify FIPS_rsa_verify_ctx FIPS_rsa_verify_digest
RSA Signature Generation based on PSS (probabilistic signature scheme)	✓		2048, 3072, 4096 bit modulus	SHA-224, SHA-256, SHA-384, SHA-512	1222 1223 1790	FIPS 186-2	FIPS_rsa_sign FIPS_rsa_sign_ctx FIPS_rsa_sign_digest
RSA Signature Verification based on PSS (probabilistic signature scheme)	✓		1024, 1536, 2048, 3072, 4096 bit modulus	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1222 1223 1790	FIPS 186-2	FIPS_rsa_verify FIPS_rsa_verify_ctx FIPS_rsa_verify_digest
ECDSA Key Pair Generation with DRBG	✓		P-224, 256, 384, 521 K- 233, 283, 409,571 B- 233, 283, 409, 571		389 390 705	FIPS 186-2	FIPS_ec_key_new FIPS_ec_key_new_by_curve_name FIPS_ec_key_check_key FIPS_ec_key_clear_flags FIPS_ec_key_copy FIPS_ec_key_dup FIPS_ec_key_get0_private_key FIPS_ec_key_get0_public_key FIPS_ec_key_get_conv_form FIPS_ec_key_set_conv_form FIPS_ec_key_get_enc_flags FIPS_ec_key_set_enc_flags FIPS_ec_key_get_flags

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

							FIPS_ec_key_set_flags FIPS_ec_key_get_key_method_data FIPS_ec_key_set_group FIPS_ec_key_insert_key_method_data FIPS_ec_key_precompute_mult FIPS_ec_key_set_asn1_flag FIPS_ec_key_set_private_key FIPS_ec_key_set_public_key FIPS_ec_key_up_ref FIPS_ec_key_new_by_curve_name FIPS_ec_key_generate_key FIPS_ec_key_set_public_key_affine_coordinates FIPS_ec_key_free FIPS_ec_point_new FIPS_ec_point_set_to_infinity FIPS_ec_point_get_affine_coordinates_gfp FIPS_ec_point_get_jprojective_coordinates_gfp FIPS_ec_point_is_at_infinity FIPS_ec_point_is_on_curve FIPS_ec_point_make_affine FIPS_ec_points_make_affine FIPS_ec_point_method_of FIPS_ec_point_mul FIPS_ec_point_free FIPS_ec_point_clear_free FIPS_ecdsa_openssl
ECDSA Signature Verification	✓		P-192,224, 256,384,521 K-163,233, 283,409,571 B-163,233, 283,409,571		389 390 705	FIPS 186-2	FIPS_ecdsa_verify FIPS_ecdsa_verify_ctx FIPS_ecdsa_verify_digest FIPS_ecdsa_sig_free
ECDSA PKV	✓		P-192, 224, 256, 384, 521 K-163, 233, 283, 409, 571 B-163,233, 283,409,571		389 390 705	FIPS 186-2	FIPS_ecdsa_verify FIPS_ecdsa_verify_ctx FIPS_ecdsa_verify_digest FIPS_ecdsa_sig_free

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

ECDSA Key Pair Generation with DRBG	✓		P-224, 256,384,521 K-233, 283,409,571 B-233, 283,409,571	Key Pair generation by testing candidates.	389 390 705	FIPS 186-4	FIPS_ec_key_new FIPS_ec_key_new_by_curve_name FIPS_ec_key_check_key FIPS_ec_key_clear_flags FIPS_ec_key_copy FIPS_ec_key_dup FIPS_ec_key_get0_private_key FIPS_ec_key_get0_public_key FIPS_ec_key_get_conv_form FIPS_ec_key_set_conv_form FIPS_ec_key_get_enc_flags FIPS_ec_key_set_enc_flags FIPS_ec_key_get_flags FIPS_ec_key_set_flags FIPS_ec_key_get_key_method_data FIPS_ec_key_set_group FIPS_ec_key_insert_key_method_data FIPS_ec_key_precompute_mult FIPS_ec_key_set_asn1_flag FIPS_ec_key_set_private_key FIPS_ec_key_set_public_key FIPS_ec_key_up_ref FIPS_ec_key_new_by_curve_name FIPS_ec_key_generate_key FIPS_ec_key_set_public_key_affine_coordinates FIPS_ec_key_free FIPS_ec_point_new FIPS_ec_point_set_to_infinity FIPS_ec_point_get_affine_coordinates_gfp FIPS_ec_point_get_jprojective_coordinates_gfp FIPS_ec_point_is_at_infinity FIPS_ec_point_is_on_curve FIPS_ec_point_make_affine FIPS_ec_points_make_affine

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

							FIPS_ec_point_method_of FIPS_ec_point_mul FIPS_ec_point_free FIPS_ec_point_clear_free FIPS_ecdsa_openssl
ECDSA Signature Generation	✓		P-224, 256, 384, 521 K- 233, 283, 409, 571 B-233, 283,409,571	SHA-224, SHA-256, SHA-384, SHA-512	389 390 705	FIPS 186-4	FIPS_ecdsa_sign FIPS_ecdsa_sign_ctx FIPS_ecdsa_sign_digest FIPS_ecdsa_sig_free
ECDSA PKV	✓		P-192, 224, 256, 384, 521 K-163, 233, 283, 409, 571 B-163,233, 283,409,571	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	389 390 705	FIPS 186-4	FIPS_ecdsa_verify FIPS_ecdsa_verify_ctx FIPS_ecdsa_verify_digest FIPS_ecdsa_sig_free
ECDSA Signature verification	✓		P-192, 224, 256, 384, 521 K-163, 233, 283, 409, 571 B-163,233, 283,409,571	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	389 390 705	FIPS 186-4	FIPS_ecdsa_verify FIPS_ecdsa_verify_ctx FIPS_ecdsa_verify_digest FIPS_ecdsa_sig_free
KEY EXCHANGE							
KASECC Component test	✓		P-224, 256,384,521 K-233, 283,409,571 B-233, 283,409,571	Section 5.7.1.2 ECC CDH Primitive	62 63 549	SP 800-56A	FIPS_ecdh_openssl FIPS_ecdh_compute_key FIPS_ec_group_new FIPS_ec_group_new_by_curve_name FIPS_ec_group_new_curve_gf2m FIPS_ec_group_new_curve_gfp FIPS_ec_group_precompute_mult FIPS_ec_group_get0_generator FIPS_ec_group_get0_seed FIPS_ec_group_get_asn1_flag

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

							FIPS_ec_group_set_asn1_flag FIPS_ec_group_get_cofactor FIPS_ec_group_get_curve_gf2m FIPS_ec_group_set_curve_gf2m FIPS_ec_group_get_curve_gfp FIPS_ec_group_set_curve_gfp FIPS_ec_group_get_curve_name FIPS_ec_group_set_curve_name FIPS_ec_group_get_degree FIPS_ec_group_set_generator FIPS_ec_group_set_point_conversion_form FIPS_ec_group_method_of FIPS_ec_key_new FIPS_ec_key_new_by_curve_name FIPS_ec_key_check_key FIPS_ec_key_clear_flags FIPS_ec_key_copy FIPS_ec_key_dup FIPS_ec_key_get0_private_key FIPS_ec_key_get0_public_key FIPS_ec_key_get_conv_form FIPS_ec_key_set_conv_form FIPS_ec_key_get_enc_flags FIPS_ec_key_set_enc_flags FIPS_ec_key_get_flags FIPS_ec_key_set_flags FIPS_ec_key_get_key_method_data FIPS_ec_key_set_group FIPS_ec_key_insert_key_method_data FIPS_ec_key_precompute_mult FIPS_ec_key_set_asn1_flag FIPS_ec_key_set_private_key FIPS_ec_key_set_public_key FIPS_ec_key_up_ref FIPS_ec_key_new_by_curve_name FIPS_ec_key_generate_key

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

							FIPS_ec_key_set_public_key_affine_coordinates FIPS_ec_key_free FIPS_ec_point_new FIPS_ec_point_set_to_infinity FIPS_ec_point_get_affine_coordinates_gfp FIPS_ec_point_get_jprojective_coordinates_gfp FIPS_ec_point_is_at_infinity FIPS_ec_point_is_on_curve FIPS_ec_point_make_affine FIPS_ec_points_make_affine FIPS_ec_point_method_of FIPS_ec_point_mul FIPS_ec_point_free FIPS_ec_point_clear_free FIPS_ec_group_clear_free
Hash Functions							
SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	✓			N/A	2039 2040 2869	FIPS 180-4	FIPS_digest FIPS_digestinit FIPS_digestupdate FIPS_digestfinal FIPS_md_ctx_init FIPS_md_ctx_create FIPS_md_ctx_copy FIPS_md_ctx_destroy FIPS_md_ctx_cleanup
Message Authentication Codes (MACs)							
HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	✓			N/A	1470 1471 2219	FIPS 198	FIPS_hmac FIPS_hmac_init FIPS_hmac_init_ex FIPS_hmac_update FIPS_hmac_ctx_copy FIPS_hmac_ctx_set_flags FIPS_hmac_final FIPS_hmac_ctx_cleanup
CMAC generate and verify with AES	✓		128,192,256 key sizes	Supports 0 length CMAC length Min 2 Max 16	2366 2367 2381 3474	SP 800-38B	FIPS_cmac_init FIPS_cmac_update FIPS_cmac_resume FIPS_cmac_final FIPS_cmac_ctx_new FIPS_cmac_ctx_copy FIPS_cmac_ctx_free FIPS_cmac_ctx_get0_cipher_ctx
CMAC generate and verify with	✓		3 key Triple-DES	Supports 0 length	1479 1480 1959	SP 800-38B	

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

Triple-DES							FIPS_cmac_ctx_cleanup
CCM	✓		128,192,256 key sizes Nonce len 7, 8, 9,10, 11, 12,13	Tag len 4,6,8,10,12,14,16	2366 2367 2381 3474	SP 800-38C	FIPS_cipherinit FIPS_cipher FIPS_cipher_ctx_new FIPS_cipher_ctx_init FIPS_cipher_ctx_ctrl FIPS_cipher_ctx_copy FIPS_cipher_ctx_set_key_length FIPS_cipher_ctx_free FIPS_cipher_ctx_cleanup
GCM encryption/ decryption	✓		128, 192, 256 bit keys 96 bit IV supported Max IV length 1024	Tag length supports 32,64,96,104,112,120,128	2366 2367 2381 3474	SP 800-38D, compliant to section 8.2.1 for IV generation	FIPS_cipherinit FIPS_cipher FIPS_cipher_ctx_new FIPS_cipher_ctx_init FIPS_cipher_ctx_ctrl FIPS_cipher_ctx_copy FIPS_cipher_ctx_set_key_length FIPS_cipher_ctx_free FIPS_cipher_ctx_cleanup
Random Number Generation							
HASH DRBG	✓		SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Prediction resistance supported for al variations		304 305 856	SP 800-90A	FIPS_drbg_init FIPS_drbg_new FIPS_drbg_instantiate FIPS_drbg_generate FIPS_drbg_get_app_data FIPS_drbg_set_app_data FIPS_drbg_set_callbacks FIPS_drbg_set_check_interval FIPS_drbg_set_rand_callbacks FIPS_drbg_set_reseed_interval FIPS_drbg_stick FIPS_drbg_get_blocklength FIPS_drbg_reseed FIPS_drbg_method FIPS_drbg_get_strength FIPS_drbg_health_check FIPS_drbg_free FIPS_drbg_uninstantiate
HMAC DRBG	✓		SHA-1, SHA-224, SHA-256,	No reseed	304 305	SP 80-90A	FIPS_drbg_init FIPS_drbg_new

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

			SHA-384, SHA-512 Prediction resistance supported for al variations		856		FIPS_drbg_instantiate FIPS_drbg_generate FIPS_drbg_get_app_data FIPS_drbg_set_app_data FIPS_drbg_set_callbacks FIPS_drbg_set_check_inte rval FIPS_drbg_set_rand_callb acks FIPS_drbg_set_reseed_int erval FIPS_drbg_stick FIPS_drbg_get_blocklength FIPS_drbg_reseed FIPS_drbg_method FIPS_drbg_get_strength FIPS_drbg_health_check FIPS_drbg_free FIPS_drbg_uninstantiate
CTR DRBG	✓		AES-192, AES-256 Prediction resistance supported for al variations	Supported derivation function for all variations	304 305 313 856	SP 800-90A	FIPS_drbg_init FIPS_drbg_new FIPS_drbg_instantiate FIPS_drbg_generate FIPS_drbg_get_app_data FIPS_drbg_set_app_data FIPS_drbg_set_callbacks FIPS_drbg_set_check_inte rval FIPS_drbg_set_rand_callb acks FIPS_drbg_set_reseed_int erval FIPS_drbg_stick FIPS_drbg_get_blocklength FIPS_drbg_reseed FIPS_drbg_method FIPS_drbg_get_strength FIPS_drbg_health_check FIPS_drbg_free FIPS_drbg_uninstantiate
Zeroization of DRBG CSPs	✓		DRBG context	Hash_DRB G HMAC_D RBG CTR_DRB G Dual_EC_ DRBG			fips_drbg_uninstantiate fips_drbg_free

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

RSA Key wrapping	✓		A service provided for calling application, but no CSPs are transported into or out of the Module	Encryption, Decryption for key wrapping 2048, 3072, 4096	IG D.9	SP 800-56B	RSA_public_encrypt RSA_private_encrypt RSA_private_decrypt RSA_public_decrypt
Other Services							
Initialization		✓	N/A	N/A	Non security function		
Self-Tests		✓	N/A	N/A	Non security function		FIPS_selftest
Get status		✓	N/A	N/A	Non security function		FIPS_module_version FIPS_module_version_text FIPS_module_mode FIPS_selftest_failed
Diffie-Hellman primitive	Shall not be used in the FIPS-Approved mode				Non-Approved	PKCS #3	

Table 4: Services

As per the SP800-131A transition on 2014-01-01, the key lengths providing less than 112 bits of security strength are listed as non-Approved functions in Table 4.1, so their usage is discouraged as they cannot be used in FIPS mode after the transition period. Also as per the SP800-131A revision1 dated November 2015, ANSIX9.31 RNG is disallowed after 2015, and it is also listed as non-Approved functions in Table 4.1 and should not be used in FIPS mode.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

Service	Roles		CSP	Modes	FIPS Approved (Cert #)	Standard
	User	CO				
RSA key generation	✓		1024, 1536, modulus size. Public key values 3, 17, and 65537.		1222 1223	FIPS 186-2
RSA signature Generation based on X9.31	✓		1024, 1536, 2048, 3072, 4096 bit modulus	SHA-1, SHA-256, SHA-384, SHA-512	1222 1223	FIPS 186-2
RSA signature Generation based on X9.31	✓		2048, 3072, 4096 bit modulus	SHA-1	1222 1223	FIPS 186-2
RSA Signature Generation based on PKCS#1 v1.5	✓		1024, 1536, bit modulus	SHA-1, SHA-256, SHA-384, SHA-512	1222 1223	FIPS 186-2
RSA Signature Generation based on PKCS#1 v1.5	✓		2048, 3072, 4096 bit modulus	SHA-1	1222 1223	FIPS 186-2
RSA Signature Generation based on PSS (probabilistic signature scheme)	✓		1024, 1536, bit modulus	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1222 1223	FIPS 186-2
RSA Signature Generation based on PSS (probabilistic signature scheme)	✓		2048, 3072, 4096 bit modulus	SHA-1	1222 1223	FIPS 186-2
RSA Key wrapping	✓		A service provided for calling application, but no CSPs are transported into or out of the Module	Encryption, Decryption for key wrapping 1024 bits	IG D.9	SP 800-56B
DSA domain parameter generation	✓		p, q, g; 1024 bits modulus size	With SHA-1 only	739 740	FIPS 186-2
DSA Key pair generation	✓		1024 bits modulus size key pair	With SHA-1 only	739 740	FIPS 186-2
DSA Signature generation	✓		1024 bits modulus size key pair	With SHA-1 only	739 740	FIPS 186-2

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

DSA domain parameter generation	✓		L=1024,N=160	With all SHA sizes	739 740	FIPS 186-4
DSA domain parameter generation	✓		L=2048,N=224,L=2048,N=256, L=3072,N=256	With SHA-1 only	739 740	FIPS 186-4
DSA Key Pair	✓		L=1024,N=160	With all SHA sizes	739 740	FIPS 186-4
DSA Key Pair	✓		L=2048,N=224,L=2048,N=256,L=3072,N=256	With SHA-1 only	739 740	FIPS 186-4
DSA Signature generation	✓		L=1024,N=160	With all SHA sizes	739 740	FIPS 186-4
DSA Signature generation	✓		L=2048,N=224,L=2048,N=256,L=3072,N=256	With SHA-1 only	739 740	FIPS 186-4
ECDSA Key Pair Generation	✓		P-192 K-163 B-163		389 390	FIPS 186-2
ECDSA Signature Generation	✓		P-192,224, 256,384,521 K-163,233, 283,409,571 B-163,233, 283,409,571		389 390	FIPS 186-2
ECDSA Key Pair Generation	✓		P-192 K-163 B-163	Key Pair generation by testing candidates.	389 390	FIPS 186-4
ECDSA Signature Generation	✓		P-192 K-163 B-163	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	389 390	FIPS 186-4
ECDSA Signature Generation	✓		P-224, 256, 384, 521 K-233, 283, 409, 571 B-233, 83,409,571	SHA-1	389 390	FIPS 186-4
KASECC Component test	✓		P-192 K-163 B-163	Section 5.7.1.2 ECC CDH Primitive	62 63	SP 800-56A
Dual EC DRBG (non-compliant)	✓		P-256 Prediction resistance supported for al variations	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	304 305	SP 800-90A
			P-384	SHA-224, SHA-		

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

			Prediction resistance supported for al variations	256, SHA-384, SHA-512		
			P-521 Prediction resistance supported for al variations	SHA-256, SHA-384, SHA-512		
RNG	✓		Seed and seed key	AES-128 AES-192 AES-256	1176 1177 1182 1388	ANS X9.31
RSA key generation with RNG	✓		2048, 3072, 4096 modulus size. Public key values 3, 17, and 65537.		1222 1223 1790	FIPS 186-4
DSA Key Pair generation with RNG	✓		L=2048,N=224,L=2048,N=256,L=3072,N=256		739 740 982	FIPS 186-4
ECDSA Key Pair Generation with RNG	✓		P-224, 256,384,521 K-233, 283,409,571 B-233, 283,409,571		389 390 705	FIPS 186-4

Table 4.1: Non Approved Services

Caveat 1: NIST SP 800-131A describes the transition associated with the use of cryptographic algorithms and key lengths. Based on the information included in this publication, the following algorithms implemented in this cryptographic module will become “disallowed” after 2013 or 2015, so their usage is discouraged as they cannot be used in FIPS mode after the transition period:

- DSA Key Generation and Digital Signature Generation with keys of length < 2048 bits.
- RSA Key Generation and Digital Signature Generation with keys of length < 2048 bits.
- EC Diffie- Hellman’s primitive using elliptic curves with keys of length < 224 bits.
- RSA Key Wrapping with keys of length < 2048 bits.
- RNG specified in ANS X9.31
- SHA-1 for digital signature generation
- HMAC with key lengths < 112 bits

Caveat 2: Elliptic Curve Diffie-Hellman (ECDH) with 163-571 bits 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) provides 80-256 bits of security strength.

Caveat 3: RSA Key Wrapping with 1024-4096 bits of keys provides 80-150 bits of security strength.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

Caveat 4: The calling application provides the entropy input to the module. There is no assurance of the minimum strength of generated keys.

Caveat 5: In case the Module's power is lost and then restored, the calling application must ensure that the keys used for the AES GCM encryption/decryption are re-distributed.

Caveat 6: Some of the above listed FIPS_* APIs call FIPS_selftest_failed() to check if the self-test was failed before providing the requested cryptographic services. The check on the self-test status is intended as an aid to the developer in preventing the accidental use of the cryptographic services while the Module is in the error state after the self-test failed. Nevertheless, this is not a guarantee that cryptographic services are absolutely not available in the error state. Sufficiently creative or unusual use of the APIs may still allow the use of some pieces of cryptographic services. It is the responsibility of the application developer to ensure that if the self-test fails during the Module initialization, the application must exit and then to initialize the Module. In case the self-test fails, the application shall not call any of the cryptographic API functions.

Caveat 7: DRBG should be used for all key generation in FIPS mode. This can be set by passing DRBG to FIPS_rand_set_method function. If RNG is used then the module will be in non-FIPS mode. The keys and CSPs should be separate and should not be shared between modes.

4.3 Operator Authentication

There is no operator authentication; assumption of role is implicit by the services that the operator invokes.

4.4 Mechanism and Authentication Strength

No authentication is required at security level 1; authentication is implicit by assumption of the role.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

5 Physical Security

This is a software module and provides no physical security.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

6 Operational Environment

The Module operates in a modifiable operational environment.

6.1 Policy

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 is single-threaded and functions entirely within the process space of the calling application. The application that uses the Module is the single user of the Module. No concurrent operators are allowed.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Cryptographic Key Management

The management of all keys/CSPs used by the Module is summarized in the table below. The key/CSP names are generic, corresponding to API parameter data structures.

Key/CSP Name	Details
128-, 192-, and 256-bit AES keys	Accessible by Roles: User, Crypto Officer Generation: by DRBG Type: Symmetric Encryption and Decryption Key Entry: API parameter Output: API parameter Storage: RAM for the lifetime of API call Zeroization APIs: FIPS_cipher_ctx_cleanup
3-Key Triple-DES Key	Accessible by Roles: User, Crypto Officer Generation: by DRBG Type: Symmetric Encryption and Decryption Key Entry: API parameter Output: API parameter Storage: RAM for the lifetime of API call Zeroization API: FIPS_cipher_ctx_cleanup
CMAC keys	Accessible by Roles: User, Crypto Officer Generation: N/A Type: 3-Key Triple key or AES key Entry: API function Output: N/A Storage: RAM for the lifetime of API call Zeroization API: CMAC_CTX_cleanup
HMAC keys	Accessible by Roles: User, Crypto Officer Generation: N/A Type: Keyed-Hash Message Authentication Entry: API function Output: N/A Storage: RAM for the lifetime of API call Zeroization API: HMAC_CTX_cleanup
HMAC key for Module integrity check	Accessible by Roles: Crypto Officer Generation: N/A Type: Keyed-Hash Message Authentication Entry: API function Output: N/A Storage: Module binary Zeroization: not required per FIPS IG 7.4.
RSA key pairs	Accessible by Roles: User, Crypto Officer Generation: DRBG Type: Asymmetric key pair Entry: API function Output: API function Storage: N/A

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

Key/CSP Name	Details
	Zeroization API: FIPS_rsa_free
DSA key pairs	Accessible by Roles: User, Crypto Officer Generation: DRBG Type: Asymmetric key pair Entry: API function Output: API function Storage: RAM for the lifetime of API call Zeroization API: FIPS_dsa_free
ECDSA key pairs	Accessible by Roles: User, Crypto Officer Generation: DRBG Type: Asymmetric key pair Entry: API function Output: API function Storage: RAM for the lifetime of API call Zeroization API: EC_KEY_free
EC Diffie-Hellman primitives	Accessible by Roles: User, Crypto Officer Generation: DRBG Type: primitives Entry: API function Output: API function Storage: RAM for the lifetime of API call Zeroization API: EC_KEY_free
RNG CSPs	Accessible by Roles: User, Crypto Officer Generation: NDRNG from the OE Type: seed and seed key Entry: API function Output: API function Storage: RAM for the lifetime of RNG instance Zeroization API: fips_rand_prng_reset
DRBG CSPs	Accessible by Roles: User, Crypto Officer Generation: NDRNG from the OE Type: V, C, HMAC Key, AES Key, and entropy input Entry: API function Output: API function Storage: RAM for the lifetime of DRBG instance Zeroization API: FIPS_drbg_free, FIPS_drbg_uninstantiate

Table 5: Key Management Details

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

7.1 Key/CSP Generation

The Module implements ANS X9.31 compliant RNG and SP 800-90A compliant DRBG services for creation of symmetric and asymmetric keys. : DRBG should be used for key generation using FIPS_rand_set_method(). If RNG is used for key generation then the module will be in non-FIPS mode. The keys and CSPs should be separate and should not be shared between modes.

The calling application is responsible for storage of generated keys returned by the Module. The seeds and entropy input are provided to the Module by the calling application. Module users (the calling application) shall use entropy sources that meet the security strength required for the random number generation mechanism: as shown in Table 2, Table 3, and Table 4 in SP 800-90A for DRBG. The entropy is supplied by means of callback functions. Those functions must return an error if the minimum entropy strength cannot be met.

7.2 Key Entry and Output

All CSPs enter the Module's logical boundary as cryptographic algorithm API parameters in plaintext. They are associated with memory locations and do not persist across power cycles. The Module does not output intermediate key generation values or other CSPs. The Module provides the resulting keys as explicit return values of key generation services to the calling application, but they do not cross the physical boundary.

7.3 Key Storage

The Module does not provide persistent key storage for keys or CSPs. The Module stores RNG and DRBG state values for the lifetime of the RNG or DRBG instance in RAM. The Module uses pointers to plaintext keys/CSPs that are passed in by the calling application. The Module does not store any CSP beyond the lifetime of an API call, with the exception of RNG and DRBG state values used for the Module's key generation services.

7.4 Key Zeroization

Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. All keys and CSPs are ephemeral and are destroyed when released by the appropriate API function calls. Keys and CSPs 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 in a single user mode by one calling application at a time to ensure that no two API functions will execute concurrently.

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.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

8 Electromagnetic Interference/Compatibility

The Module's electromagnetic interference (EMI) and electromagnetic compatibility (EMC) features are summarized in Table 6: EMI and EMC.

Testing Platform	Product Name/Model	Model Number	EMI/EMC Notes
oMG	oMG	2000	Compliant to FCC part 15 Class A per FCC report
Dell	PowerEdge	R210	Compliant to FCC part 15 Class A per "PowerEdge R210 Dell Technical Guide"
HP	HP Pro Book	640 G1	Compliance to FCC Part 15 Class B per "HP ProBook 640 G1 Notebook PC QuickSpecs"

Table 6: EMI and EMC

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

9 Self-Tests

The module performs all the power-up self test upon initialization of the module and before the module becomes usable. All the tests are performed automatically without requiring any operator intervention. Call to the FIPS_selftest() function is made in the constructor which performs the test. The invocation of FIPS_selftest() function performs all power-up self-tests listed below in section 9.2 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.

9.1 Integrity test

During the software build process, the Module is used to compute a HMAC-SHA-1 message authentication code (MAC) of the Module binary—the MAC and the required key are then stored with the Module. Prior to loading the Module, a HMAC-SHA-1 MAC of the binary is again computed and compared to the original. If the comparison passes, the Module is loaded and the power-up self-tests are run; if the self- tests pass, the Module enters FIPS-Approved mode. If the comparison for integrity check fails, the FIPS_selftest()function returns "0" to indicate the failure and set a global flag for the failing status. The calling application shall check the Module status before proceeding with any further action.

9.2 Power-up Tests

At Module start-up, the following Known Answer Tests (KAT) or Pair-wise Consistency Test (PCT) in place of KAT for DSA and ECDSA are performed by FIPS_selftest():

- AES separate encrypt and decrypt, ECB mode, 128 bit key
- AES CCM separate encrypt and decrypt, 192 key length
- AES GCM separate encrypt and decrypt, 256 key length
- XTS-AES 128- or 256-bit key size to support XTS-AES-128 or XTS-AES-256 respectively
- AES CMAC generate and verify CBC mode, 128, 192, 256 key lengths
- Triple-DES separate encrypt and decrypt, ECB mode, 3-Key
- Triple-DES CMAC generate and verify, CBC mode, 3-Key
- HMAC one KAT per SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512
- SHA KATs is performed via HMAC KATs (allowed via FIPS 140-2 IG 9.1)
- ANS X9.31 RNG 128-, 192-, 256-bit AES keys
- DRBG 800-90A
    - CTR_DRBG: AES, 256-bit with and without derivation function
    - HASH_DRBG: SHA-256
    - HMAC_DRBG: SHA-256
    - Dual_EC_DRBG: P-256 and SHA-256
- RSA sign and RSA verify separately using 2048-bit key, SHA-256
- DSA PCT on signing and verifying signature using 2048-bit key, SHA-384

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

- ECDSA PCT on key generation, signing, and verifying using P-224, K-233 and SHA-512
- ECC CDH shared secret calculation per section 5.7.1.2 of SP 800-56A, IG 9.6

Depending on whether the underlying Operational Environment has AES-NI capable processor with this feature enabled, the KATs for AES and all algorithms that rely on AES (i.e., AES CCM/GCM/CMAC/XTS, RNG and CTR_DRBG) may or may not utilize the AES-NI support from the processor. When the AES-NI support is enabled, the AES implementation utilizes the AES-NI support from the processor. When the AES-NI support is disabled, the AES implementation is solely in software. In both AES-NI enabled and disabled scenarios, there is one and only one AES implementation in the Module is used. The KATs of AES and algorithms relying on AES as well as the subsequent calls to these cryptographic services consistently use the same AES implementation.

The Module has been tested on the Dell PowerEdge R210 platform containing Intel x86 processor with and without the AES-NI support enabled. AES and all algorithms that rely on AES also have algorithm certificates for Intel x86 processor with and without the AES-NI support enabled.

9.3 Conditional Tests

The Module also implements the following conditional tests:

- ANS X9.31 RNG continuous test
- DRBG SP 800-90 continuous test
- RSA pairwise consistency test on each generation of a key pair
    - Use private key for signature generation and public key for signature verification
    - Use public key for key encryption and private key for key decryption
- DSA pairwise consistency test on each generation of a key pair
    - Use private key for signature generation and public key for signature verification
- ECDSA pairwise consistency test on each generation of a key pair
    - Use private key for signature generation and public key for signature verification

9.4 On-Demand Self-Test

On-demand self-tests can be invoked by calling FIPS_selftest() function. This function performs all the power-up self-tests listed in section 9.2. The function returns “1” on successful completion of all the tests and returns “0” if any error occurs.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

10 Design Assurance

10.1 Configuration Management

CDs containing the uncompressed and expanded contents of the source code distribution *openssl-fips-2.0.1.tar.gz* are obtained from the OpenSSL Software Foundation (OSF). The openssl-fips-2.0.1 source code is compiled and built for the Dell PowerEdge R210, oMG 2000 and HP Pro Book 640 G1 platforms. The generated resulting object code is in file fipscanister.o. The object code file is distributed in the packages as described in section 2.3.

Upon receiving the CDs from OSF, the source code for the Module is then stored on a server that is connected to a private corporate intranet. Changes to the source code, and other required files, are managed with the git distributed version control system, which provides traceability between developers, the source code, and the released binary module. Each binary is tracked with an embedded build number that has a matching tag in the revision control system, which identifies the source files that were used to produce the binary.

10.2 Installation and Usage Guidance

The Module is a monolithic FIPS Object Module. It is designed for indirect use via the OpenSSL API. For the convenience of use, Nokia delivers the OpenSSL library (i.e., libcrypto.so) with the Module version 2.1 embedded within as part of the OpenSSL build process. The applications can then link to the OpenSSL library to utilize the FIPS validated cryptographic functions provided by the embedded fipscanister object code. For Module version 2.2 it is distributed as a library file fipscanister.lib.

Developers who intend to build FIPS capable OpenSSL library by combining the FIPS validated canister object code and a version of the OpenSSL product that is suitable for use with this object module shall follow two steps listed below:

1. The HMAC-SHA-1 digest of the Module object file must be calculated and verified against the pre-calculated digest to ensure the integrity of the Module object file.
2. A HMAC-SHA-1 digest of the Module must be generated and embedded in the Module for use by the FIPS_check_incore_fingerprint() function at runtime initialization.

The fips_standalone_sha1 command can be used to perform the verification of the Module object file and to generate the new HMAC-SHA-1 digest for the runtime executable object. Failure to embed the digest in the executable object will prevent initialization of the Module into the FIPS-Approved mode.

At runtime, the FIPS_check_incore_fingerprint() function compares the embedded HMAC-SHA-1 digest with a digest generated form the Module object code.

The calling application interfacing with the Module is outside the cryptographic boundary, but it has to link the Module appropriately by following the steps described above in order to ensure that the Module is compliant with FIPS 140-2.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

11 Mitigation of Other Attacks

No other attacks are mitigated.

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

12 Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
FSM	Finite State Model
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OE	Operational Environment
OFB	Output Feedback
O/S	Operating System
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SLA	Service Level Agreement
SOF	Strength of Function
SSH	Secure Shell
SVT	Scenario Verification Testing
TDES	Triple DES
UI	User Interface

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

13 References

[1] FIPS 140-2 Standard, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[2] FIPS 140-2 Implementation Guidance, http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

[3] FIPS 140-2 Derived Test Requirements, http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf

[4] FIPS 197, Advanced Encryption Standard (AES), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[5] FIPS 180-4 Secure Hash Standard, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[7] FIPS 186-2, Digital Signature Standard, http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf

[8] FIPS 186-3 Digital Signature Standard (DSS), http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[9] ANS X9.31 Appendix A.2.4, Random Number Generator, http://csrc.nist.gov/groups/STM/cavp/documents/rng/931rngext.pdf

[10] NIST SP 800-67 Revision 1, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf

[11] NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

[12] NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf

[13] NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[14] NIST SP 800-38E, Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

[15] NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes using Discrete Logarithm Cryptography (Revised), http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.

[16] NIST SP 800-56B, Recommendation for Pair-Wise Establishment Schemes Using Integer Factorization Cryptography, http://csrc.nist.gov/publications/nistpubs/800-56B/sp800-56B.pdf

[17] NIST SP 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

[18] NIST SP 800-131A Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

©2018 Nokia of America Corporation (NoAC). This document can be reproduced and distributed only whole and intact, including this copyright notice.