ORACLE®
Linux

FIPS 140-2 Non-Proprietary Security Policy

Oracle Linux OpenSSL Cryptographic Module

FIPS 140-2 Level 1 Validation

Software Version: R7-3.0.0 and R7-4.0.0

Date: March 18th, 2020

Title: Oracle Linux OpenSSL Cryptographic Module Security Policy

Date: March 18th, 2020

Author: Oracle Security Evaluations – Global Product Security

Contributing Authors:

Oracle Linux Engineering

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

TABLE OF CONTENTS

Section **Title** **Page**

List of Tables

List of Figures

1. Introduction

1.1 Overview

Oracle Linux is a set of cryptographic libraries, services, and user level cryptographic applications that are validated at FIPS 140-2 level 1, providing a secure foundation for vendor use in developing dependent services, applications, and even purpose built appliances that may be FIPS 140-2 validated.

This section is informative to the reader to reference other cryptographic services of Oracle Linux. Only the software listed in section 3.1 is subject to the FIPS 140-2 validation. The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when supported if the specific operational environment is not listed on the validation certificate.

The FIPS 140-2 validation is performed at Security Level 1, on software only modules that do not make any claims about the hardware enclosure. This allows vendors to develop their own modules using these basic cryptographic modules and validate the new modules at a higher FIPS 140-2 security level.

The following cryptographic modules are included in Oracle Linux:

- OpenSSL– a software cryptographic module supporting FIPS 140-2-approved cryptographic algorithms for protocols like TLS 1.2, OpenSSH, and HTTPS
- OpenSSH-Server – supplies cryptographic support for the SSH protocol
- OpenSSH-Client – supplies cryptographic support for the SSH protocol
- NSS Softokn Cryptographic Module – supplies cryptographic support for TLS, PKCS #5, PKCS #7, PKCS #11 (version 2.20), PKCS #12, S/MIME, X.509 v3 certificates, and other security standards supporting FIPS 140-2 validated cryptographic algorithms
- Oracle Linux Unbreakable Enterprise Kernel (UEK) – a software only cryptographic module via the Kernel Crypto API that is an optimized kernel with a wide range of advanced features and improvements for enterprise workloads. The UEK provides general-purpose cryptographic services and block storage encryption.
- GnuTLS – general purpose cryptographic module to support TLS network protocols
- Libgcrypt – supplies general cryptographic support for GPG.
- Libreswan – provides the IKE protocol version 2 key agreement services required for IPSEC.

This Security Policy describes the features and design of the Oracle Linux OpenSSL Cryptographic Module using the terminology contained in the FIPS 140-2 specification. FIPS 140-2, Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CSE Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-2. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

1.2 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Oracle Linux OpenSSL Cryptographic Module Non-Proprietary Security Policy
- Other supporting documentation as additional references

With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Oracle and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Oracle.

2. Oracle Linux OpenSSL Cryptographic Module

2.1 Functional Overview

The Oracle Linux OpenSSL Cryptographic Module (hereafter referred to as the "Module") is a software module supporting FIPS 140-2 Approved cryptographic algorithms within Oracle Linux. The code base of the Module is formed in a combination of standard OpenSSL shared Library, OpenSSL FIPS Object Module, and development work by Oracle Linux engineering. The scope of testing for this validation covers versions R7-3.0.0 running Oracle Linux 7.5 and R7-4.0.0 running on Oracle Linux 7.6. The Oracle Linux OpenSSL Module is distributed with an open-source distribution. The Module provides a C language application program interface (API) for use by other processes that require cryptographic functionality.

Oracle Linux OpenSSL supports following three types of cryptographic implementations. The implementations available for an algorithm are listed in Table 2 and they can be selected based on the environment variable OPENSSL_ia32cap. Please refer to its man page for the details.

a) OpenSSL in Native C Programming Language;
b) AES-NI hardware acceleration for X86 processors;
c) Assembler implementation.

2.2 FIPS 140-2 Validation Scope

The following table shows the security level for each of the eleven sections of the validation. See Table 1 below.

Security Requirements Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles and Services and Authentication	1
Finite State Machine Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	1

Table 1: FIPS 140-2 Security Requirements

3. Cryptographic Module Specification

3.1 Definition of the Cryptographic Module

The Oracle Linux OpenSSL Module is defined as a multi-chip standalone module as defined by the requirements within FIPS PUB 140-2. The logical cryptographic boundary of the module consists of shared library files and their integrity check HMAC files, which are delivered through the Oracle Linux Yum Server as listed below:

The module version R7-3.0.0 was tested on Oracle Linux 7.5 and is contained within the RPM file with version openssl-libs-1.0.2k-12.0.3.el7.x86_64.rpm, which contains the following files:

- /usr/lib64/.libcrypto.so.1.0.2k.hmac (64 bits)
- /usr/lib64/.libssl.so.1.0.2k.hmac (64 bits)
- /usr/lib64/libcrypto.so.1.0.2k (64 bits)
- /usr/lib64/libssl.so.1.0.2k (64 bits)

The module instantiation for version R7-3.0.0 for Oracle Linux 7.5 is provided by the dracut-fips package with the version of the RPM file of dracut-fips-033-535.0.5.el7_5.1.x86_64.rpm.

The AES-NI configuration of the UEK kernel for module version R7-3.0.0 for Oracle Linux 7.5 is provided by the dracut-fips package with the version of the RPM file of dracut-fips-aesni-033-535.0.5.el7_5.1.x86_64.rpm

The module version R7-4.0.0 was tested on Oracle Linux 7.6 and is contained within the RPM file with version openssl-libs-1.0.2k-16.0.1.el7.x86_64.rpm, which contains the following files:

- /usr/lib64/.libcrypto.so.1.0.2k.hmac (64 bits)
- /usr/lib64/.libssl.so.1.0.2k.hmac (64 bits)
- /usr/lib64/libcrypto.so.1.0.2k (64 bits)
- /usr/lib64/libssl.so.1.0.2k (64 bits)

The module instantiation for version R7-4.0.0 is provided by the dracut-fips package with the version of the RPM file of dracut-fips-033-554.0.3.el7.x86_64.rpm.

The AES-NI configuration of the UEK kernel for module version R7-4.0.0 for Oracle Linux 7.6 is provided by the dracut-fips package with the version of the RPM file of dracut-fips-aesni-033-554.0.3.el7.x86_64.rpm

The Oracle OpenSSL package includes the binary files, integrity check HMAC files, Man Pages and the OpenSSL engines provided by the standard OpenSSL shared library. The OpenSSL engines and their shared object files are not part of the Module, and therefore they must not be used when operating the Module.

The Module shall be instantiated by the dracut-fips package with the RPM file version specified above. The dracut-fips RPM package is only used for the installation and instantiation of the Module. This code is not active when the Module is operational and does not provide any services to users interacting with the Module. Therefore the dracut-fips RPM package is outside the Modules' logical boundary.

Figure 1 shows the logical block diagram of the module executing in memory on the host system.

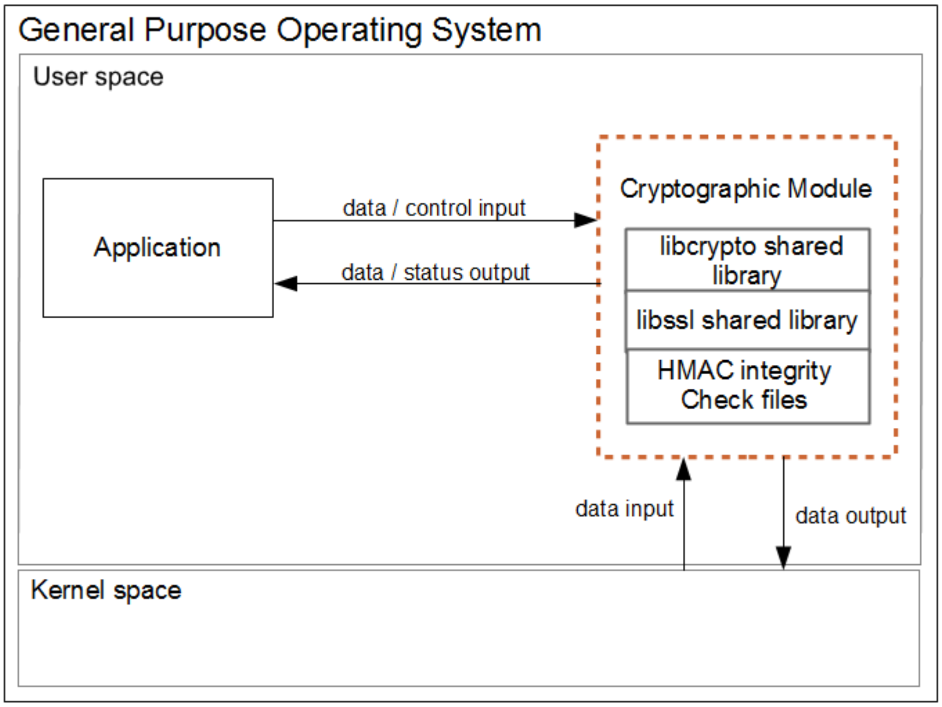

Figure 1: Oracle Linux OpenSSL Logical Cryptographic Boundary

3.2 Definition of the Physical Cryptographic Boundary

The physical cryptographic boundary of the module is defined as the hard enclosure of the host system on which it runs. See figure 2 below. No components are excluded from the requirements of FIPS PUB 140-2.

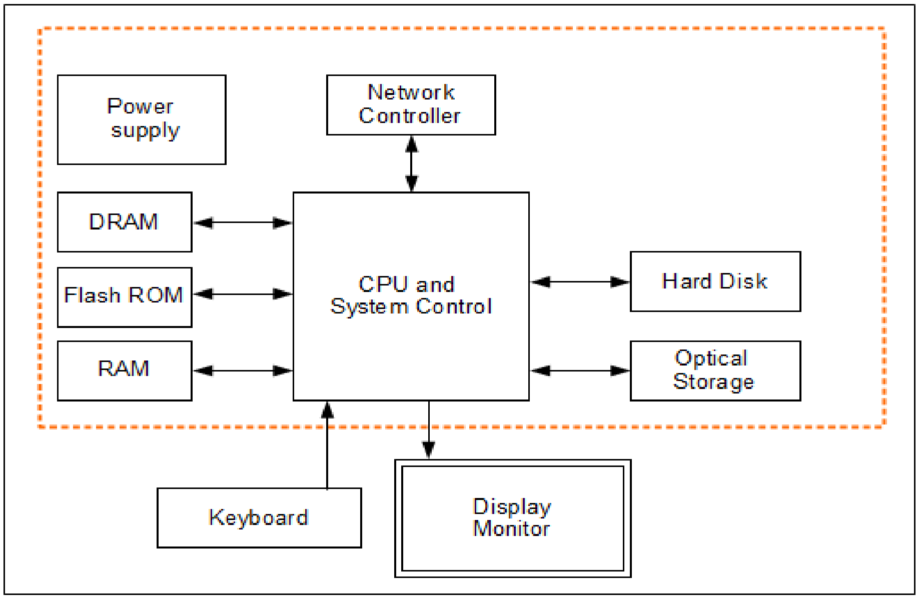

Figure 2: Oracle Linux OpenSSL Hardware Block Diagram

3.3 Approved or Allowed Security Functions

The Oracle Linux OpenSSL Cryptographic Module contains the following FIPS Approved Algorithms listed in Table 2. Certificate Numbers listed in column ‘OL7.5’ corresponds to module version R7-3.0.0 and certificate numbers listed in column OL 7.6 corresponds to module version R7-4.0.0.

Once the module is operational, the mode of operation is implicitly assumed depending on the services/security function invoked. By default the module enters Approved mode after the power-up tests succeed. In Approved mode, only approved or allowed security functions can be used as specified in table 2 and 3 and services listed in table 8.

Approved or Allowed Security Functions		OL 7.5 Cert #	OL 7.6 Cert #
Symmetric Algorithms			
AES	**AES-NI Implementation :** **ECB**; **CBC; CFB1;CFB8; CFB128; OFB** (e/d; 128 , 192 , 256); **CTR** (int only; 128 , 192 , 256) **CCM** (KS: 128 , 192 , 256) (Assoc. Data Len Range: 0 - 0 , 2^16) (Payload Length Range: 0 - 32 (Nonce Length(s): 7 8 9 10 11 12 13 (Tag Length(s): 4 6 8 10 12 14 16) **CMAC** (Generation/Verification) (KS: 128; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 192; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 256; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) **GCM** (KS: AES_128(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_192(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_256(e/d) Tag Length(s): 128 120 112 104 96 64 32) IV Generated: (Internal (using Section 8.2.1)) ; PT Lengths Tested: (0 , 128 , 1024 , 120 , 248) ; AAD Lengths tested: (0 , 128 , 1024 , 120 , 248) ; 96BitIV_Supported ; OtherIVLen_Supported GMAC_Supported **KW** (AE , AD , AES-128 , AES-192 , AES-256 , FWD , 128 , 256 , 192 , 320 , 4096) **KWP** (AE , AD , AES-128 , AES-192 , AES-256 , FWD , 8 , 72 , 32 , 96 , 808)	C 117	C 429
	VPAES SSSE3 Implementation: **ECB**; **CBC; CFB1;CFB8; CFB128; OFB** (e/d; 128 , 192 , 256); **CTR** (int only; 128 , 192 , 256) **CCM** (KS: 128 , 192 , 256) (Assoc. Data Len Range: 0 - 0 , 2^16) (Payload Length Range: 0 - 32 (Nonce Length(s): 7 8 9 10 11 12 13 (Tag Length(s): 4 6 8 10 12 14 16) **CMAC** (Generation/Verification) (KS: 128; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 192; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 256; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) **GCM** (KS: AES_128(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_192(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_256(e/d) Tag Length(s): 128 120 112 104 96 64 32)	C 118	C 422

Approved or Allowed Security Functions		OL 7.5 Cert #	OL 7.6 Cert #
	IV Generated: (Internal (using Section 8.2.1)) ; PT Lengths Tested: (0 , 128 , 1024 , 120 , 248) ; AAD Lengths tested: (0 , 128 , 1024 , 120 , 248) ; 96BitIV_Supported ; OtherIVLen_Supported GMAC_Supported **KW** (AE , AD , AES-128 , AES-192 , AES-256 , FWD , 128 , 256 , 192 , 320 , 4096) **KWP** (AE , AD , AES-128 , AES-192 , 256 , FWD , 8 , 72 , 32 , 96 , 808)		
	AES – Straight Assembler Implementation: **ECB**; **CBC; CFB1;CFB8; CFB128; OFB** (e/d; 128 , 192 , 256); **CTR** (int only; 128 , 192 , 256) **CCM** (KS: 128 , 192 , 256) (Assoc. Data Len Range: 0 - 0 , 2^16) (Payload Length Range: 0 - 32 (Nonce Length(s): 7 8 9 10 11 12 13 (Tag Length(s): 4 6 8 10 12 14 16) **CMAC** (Generation/Verification) (KS: 128; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 192; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) (KS: 256; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 16) **GCM** (KS: AES_128(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_192(e/d) Tag Length(s): 128 120 112 104 96 64 32) (KS: AES_256(e/d) Tag Length(s): 128 120 112 104 96 64 32) IV Generated: (Internal (using Section 8.2.1)) ; PT Lengths Tested: (0 , 128 , 1024 , 120 , 248) ; AAD Lengths tested: (0 , 128 , 1024 , 120 , 248) ; 96BitIV_Supported ; OtherIVLen_Supported GMAC_Supported **KW** (AE , AD , AES-128 , AES-192 , AES-256 , FWD , 128 , 256 , 192 , 320 , 4096) **KWP** (AE , AD , AES-128 , AES-192 , 256 , FWD , 8 , 72 , 32 , 96 , 808)	C 119	C 423
Triple DES	**C Implementation:** TECB(KO 1 e/d,) ; TCBC(KO 1 e/d,) ; TCFB1(KO 1 e/d,) ; TCFB8(KO 1 e/d,) ; TCFB64(KO 1 e/d,) ; OFB(KO 1 e/d,) ; CTR (int only)	C 117	C 429
	C Implementation: **CMAC** (KS: 3-Key; Generation/Verification; Block Size(s): Full / Partial ; Msg Len(s) Min: 0 Max: 2^16 ; Tag Len(s) Min: 0 Max: 8)	C 117 C 118 C 119	C 422 C 423 C 429
Secure Hash Standard (SHS)			
SHS	**AESNI, SHA1 AVX, SHA2 ASM:** **SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only)	C 117	C 429
	Assembler Implementation: **SHA-1** (BYTE-only)	C 118	C 422
	SSSE3 Implementation: **SHA-1** (BYTE-only)	C 119	C 423
Data Authentication Code			

Approved or Allowed Security Functions		OL 7.5 Cert #	OL 7.6 Cert #
HMAC	**using AESNI, SHA1 AVX, SHA2 ASM:** **HMAC-SHA1** (Key Sizes Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA224** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA256** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA384** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA512** (Key Size Ranges Tested: KS<BS KS=BS KS>BS)	C 117	C 429
	Using SHA1 assembler Implementation: **HMAC-SHA1** (Key Size Ranges Tested: KS<BS KS=BS KS>BS)	C 118	C 422
	using SHA1 SSSE3 Implementation: **HMAC-SHA1** (Key Size Ranges Tested: KS<BS KS=BS KS>BS)	C 119	C 423
RSA	**C Implementation:** **FIPS186-2:** **ALG[ANSI X9.31**]: SIG(gen); 4096 SHS: SHA-256, SHA-384, SHA-512 **ALG[RSA SSA-PKCS1_V1_5]:** SIG(gen) 4096 , SHS: SHA-224, SHA-256, SHA-384 , SHA-512 **ALG[RSA SSA-PSS]:** SIG(gen); 4096 , SHS: SHA-224 , SHA-256, SHA-384 , SHA-512 **FIPS186-4:** **186-4 KEY(gen**): FIPS186-4_Random_e **PGM (ProbRandom:** (2048 , 3072) **PPTT:**(C.3) **ALG [ANSI X9.31]** Sig(Gen): (2048 SHA(256 , 384 , 512)) (3072 SHA(256 , 384 , 512)) Sig(Ver): (1024 SHA(1 , 256 , 384 , 512)) (2048 SHA(1 , 256 , 384 , 512)) (3072 SHA(1 , 256 , 384 , 512)) **ALG[RSA SSA-PKCS1_V1_5]** SIG(gen) (2048 SHA(224 , 256 , 384 , 512)) (3072 SHA(224 , 256 , 384 , 512)) SIG(Ver) (1024 SHA(1 , 224 , 256 , 384 , 512)) (2048 SHA(1 , 224 , 256 , 384 , 512)) (3072 SHA(1 , 224 , 256 , 384 , 512)) **[RSA SSA-PSS]:** Sig(Gen): (2048 SHA(224 , 256 , 384 , 512)) (3072 SHA(224 , 256 , 384 , 512) Sig(Ver): (1024 SHA(1 , 224 , 256 , 384 , 512)) (2048 SHA(1 , 224 , 256 , 384 , 512)) (3072 SHA(1 , 224 , 256 , 384 , 512))	C 117	C 429
DSA	**C Implementation:** **FIPS186-4:** **PQG (gen) PARMS TESTED**: [(2048, 224)SHA(224 , 256 , 384 , 512); (2048,256)SHA(256 , 384 , 512); (3072,256) SHA(256 , 384 , 512)] **PQG (ver) PARMS TESTED:** [(1024,160) SHA(1 , 224 , 256 , 384 , 512); (2048,224) SHA(224 , 256 , 384 , 512); (2048,256) SHA(256 , 384 , 512); (3072,256) SHA(256 , 384 , 512)] **KeyPairGen:** [(2048,224) ; (2048,256) ; (3072,256)] **SIG (gen) PARMS TESTED:** [(2048,224) SHA(224 , 256 , 384 , 512); (2048,256) SHA(224 , 256 , 384 , 512); (3072,256) SHA(224 , 256 , 384 , 512);] **SIG (ver) PARMS TESTED:** [(1024,160) SHA(1 , 224 , 256 , 384 , 512); (2048,224) SHA(1 , 224 , 256 , 384 , 512); (2048,256) SHA(1 , 224 , 256 , 384 , 512); (3072,256) SHA(1 , 224 , 256 , 384 , 512)]	C 117	C 429
ECDSA	**C Implementation:**	C 117	C 429

Approved or Allowed Security Functions		OL 7.5 Cert #	OL 7.6 Cert #
	FIPS186-4: **KPG:** CURVES(P-256 P-384 P-521 Testing Candidates) **PKV:** CURVES(P-256 P-384 P-521) **SigGen:** CURVES(P-256: (SHA-224, 256, 384, 512) P-384: (SHA-224, 256, 384, 512) P-521: (SHA-224, 256, 384, 512) **SigVer:** CURVES(P-256: (SHA-1, 224, 256, 384, 512) P-384: (SHA-1, 224, 256, 384, 512) P-521: (SHA-1, 224, 256, 384, 512))		
Random Number Generation			
DRBG	**C Implementation using AESNI, SHA1 AVX, SHA2 ASM:** **Hash_Based DRBG:** Prediction Resistance Tested: Enabled and Not Enabled (SHA-1 , SHA-224 , SHA-256 , SHA-384 , SHA-512) **HMAC_Based DRBG:** Prediction Resistance Tested: Enabled and Not Enabled (SHA-1 , SHA-224 , SHA-256 , SHA-384 , SHA-512) **CTR_DRBG:** Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128 , AES-192 , AES-256) BlockCipher_No_df: (AES-128 , AES-192 , AES-256)	C 117	C 429
	Using AES and SHA1 assembler: **Hash_Based DRBG**: Prediction Resistance Tested: Enabled and Not Enabled (SHA-1) **HMAC_Based DRBG:** Prediction Resistance Tested: Enabled and Not Enabled (SHA-1) **CTR_DRBG**: Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128 , AES-192 , AES-256) BlockCipher_No_df: (AES-128 , AES-192 , AES-256)	C 118	C 422
	Using VPAES and SHA1 SSSE3: **Hash_Based DRBG**: Prediction Resistance Tested: Enabled and Not Enabled (SHA-1) **HMAC_Based DRBG:** Prediction Resistance Tested: Enabled and Not Enabled (SHA-1) **CTR_DRBG:** Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128 , AES-192 , AES-256) BlockCipher_No_df: (AES-128 , AES-192 , AES-256)	C 119	C 423
Key Establishment (All of NIST SP 800-56A Except KDF)			
Diffie-Hellman	**C Implementation:** **FFC:** (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Full Validation Key Regeneration) **SCHEMES: Ephem**: (KARole: Initiator / Responder) FB FC DSA	CVL C 117	CVL C 429
EC Diffie-Hellman	**C Implementation:** **ECC**: (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Full Validation Key Regeneration) **SCHEMES**: **EphemUnified**: (KARole: Initiator / Responder) EC: P-256 ED: P-384 EE: P-521		
Key Derivation (NIST SP 800-135 Section 4.2 in TLS 1.0, 1.1, and 1.2)			
TLS	**C Implementation:** (TLS1.0/1.1, TLS 1.2 (SHA 256, 384)	CVL C 117	CVL C 429

Table 2: FIPS Approved or Allowed Security Functions

3.4 Non-Approved But Allowed Security Functions

The following are considered non-Approved but allowed security functions provided by the Module:

Algorithm	Usage
RSA Key Wrapping	Key wrapping, key establishment methodology provides between 112 and 256 bits of encryption strength, non-compliant less than 112 bits.
Diffie-Hellman (CVL Certs. #C 117 and #C 429)	Key agreement, key establishment methodology provides between 112 and 256 bits of encryption strength, non-compliant less than 112 bits.
EC Diffie-Hellman (CVL Certs. #C 117 and #C 429)	Key agreement, key establishment methodology provides between 112 and 256 bits of encryption strength, non-compliant less than 112 bits.
NDRNG	Used for seeding NIST SP 800-90A DRBG.
MD5	Message digest used in TLS only

Table 3: Non-Approved but Allowed Security Functions

3.5 Non-Approved Security Functions

Security functions listed in the table below, make use of non-approved cryptographic algorithms. Use of any of these algorithms and services in Table 9 will put the module in the non-Approved mode implicitly. The services associated with these algorithms are specified in section 7.2.

Algorithm	Usage
ANSI X9.31 RNG	Random Number Generation
Camellia	Encryption/Decryption
CAST	Encryption/Decryption
DES	Encryption/Decryption
Diffie-Hellman	Key agreement using keys less than 2048 bits
DSA	Parameter /Key generation/Signature generation with keys not listed in Table 2
EC Diffie-Hellman	Key agreement using NIST curve P-192
ECDSA	Key generation/Signature generation with NIST curve P-192
IDEA	Encryption/Decryption
J-PAKE	Password Authenticated Key Exchange
MD2	Hash Function
MD4	Hash Function
MDC2	Hash Function
RC2	Encryption/Decryption
RC4	Encryption/Decryption
RC5	Encryption/Decryption
RIPEMD	Hash Function
RSA	Key generation/Signature generation: keys less than 2048 bits
SHA-1	Signature generation
Whirlpool	Hash Function

Table 4: Non-Approved Disallowed Functions

4. Module Ports and Interfaces

The module interfaces can be categorized as follows:

- Data Input Interface
- Data Output Interface
- Control Input interface
- Status Output Interface

The module can be accessed by utilizing the API function it provides. Table 5 below, shows the mapping of interfaces as per FIPS 140-2 Standard.

FIPS 140 Interface	Module Interfaces
Data Input	API Input Parameters
Data Output	API Output Parameters
Control Input	API Function Calls
Status Output	API Return Values, error message

Table 5: Mapping of FIPS 140 Logical Interfaces

5. Physical Security

The Module is comprised of software only and thus does not claim any physical security.

6. Operational Environment

6.1 Tested Environments

The Modules were tested on the following environments with and without PAA i.e. AES-NI:

Module Version	Operating Environment	Processor	Hardware
R7-3.0.0	Oracle Linux 7.5 64 bit	Intel® Xeon® Silver 4114	Oracle Server X7-2
R7-4.0.0	Oracle Linux 7.6 64 bit	Intel® Xeon® Silver 4114	Oracle Server X7-2
R7-4.0.0	Oracle Linux 7.6 64 bit	AMD® EPYC® 7551	Oracle Server X7-2

Table 6: Tested Operating Environment

6.2 Vendor Affirmed Environments

The following platforms have not been tested as part of the FIPS 140-2 level 1 certification however Oracle “vendor affirms” that these platforms are equivalent to the tested and validated platforms. Additionally, Oracle affirms that the module will function the same way and provide the same security services on any of the systems listed below.

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600/E5-2600 v2	Cisco UCS B200 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Cisco UCS B200 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Cisco UCS B200 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2400/E5-2400 v2	Cisco UCS B22 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-2800/E7-8800	Cisco UCS B230 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-2800/E7-8800 v3	Cisco UCS B260 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600/E5-4600 v2	Cisco UCS B420 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v3 & v4	Cisco UCS B420 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-2800/E7-8800	Cisco UCS B440 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-2800 v2/E7-4800 v2/E7-8800 v2/E7-4800 v3/E7-8800 v3	Cisco UCS B460 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Cisco UCS B480 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2400/E5-2400 v2	Cisco UCS C22 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600/E5-2600 v2	Cisco UCS C220 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Cisco UCS C220 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Cisco UCS C220 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2400/E5-2400 v2	Cisco UCS C24 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600/E5-2600 v2	Cisco UCS C240 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Cisco UCS C240 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Cisco UCS C240 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-2800 v2/E7-4800 v2, v3 & v4/E7-8800 v2 & v4	Cisco UCS C460 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Cisco UCS C480 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® D-1500	Cisco UCS E1120D-M3/K9
Oracle Linux 7.6 64-bit	Intel® Xeon® D-1500	Cisco UCS E180D-M3/K9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge FC630
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v3	Dell PowerEdge FC830
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge M630 Blade

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v4	Dell PowerEdge M830 Blade
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge R630
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge R730
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge R730xd
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4	Dell PowerEdge R930
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge T630
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2/E7-8800 v2	Fujitsu PRIMEQUEST 2400E
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2400E2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2400E3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2	Fujitsu PRIMEQUEST2400L
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST2400L2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2400L3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2	Fujitsu PRIMEQUEST 2400S
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2	Fujitsu PRIMEQUEST 2400S Lite
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2400S2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2400S2 Lite
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2400S3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2400S3 Lite
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Fujitsu PRIMEQUEST 2800B
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2800B2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2800B3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Fujitsu PRIMEQUEST 2800E
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2800E2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQUEST 2800E3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Fujitsu PRIMEQUEST 2800L
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Fujitsu PRIMEQUEST 2800L2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v4	Fujitsu PRIMEQEST 2800L3
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Fujitsu PRIMEQUEST 3800B
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Fujitsu PRIMERGY BX2580 M1
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Fujitsu PRIMERGY BX2580 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Fujitsu PRIMERGY CX2560 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Fujitsu PRIMERGY RX2530 M1
Oracle Linux 7.6 64-bit	Intel® Xeon®E5-2600 v4	Fujitsu PRIMERGY RX2530 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Fujitsu PRIMERGY RX2530 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Fujitsu PRIMEGY RX2540 M1
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Fujitsu PRIMERGY RX2540 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Fujitsu PRIMERGY RX2540 M4
Oracle Linux 7.6 64-bit	Intel® Xeon®E7-4800 v2/E7-8800 v2	Fujitsu PRIMERGY RX4770 M1
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3/E7-8800 v3	Fujitsu PRIMERGY RX4770 M2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	Fujitsu PRIMERGY RX4770 M3
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Fujitsu PRIMERGY RX4770 M4

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Hitachi BladeSymphony BS2500 HC0A1
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi BladeSymphony BS2500 HE0A2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3/E7-8800 v3	Hitachi BladeSymphony BS2500 HE0E2
Oracle Linux 7.6 64-bit	Intel® Xeon®E5-2600 v3	Hitachi BladeSymphony BS500 BS520H B3
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3/E7-8800 v3	Hitachi BladeSymphony BS500 BS520X B2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Hitachi Compute Blade 2500 CB520H B3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi Compute Blade 2500 CB520H B4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Hitachi Compute Blade 2500 CB520X B2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Hitachi Compute Blade 2500 CB520X B3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Hitachi Compute Blade 500 CB520H B3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi Compute Blade 500 CB520H B4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Hitachi Compute Blade 500 CB520X B2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi HA8000 RS210 AN2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi HA8000 RS220 AN2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Hitachi QuantaGrid D51B-2U
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Hitachi QuantaPlex T41S-2U
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Hitachi Vantara Hitachi Advanced Server DS120
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Hitachi Vantara Hitachi Advanced Server DS220
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Hitachi Vantara Hitachi Advanced Server DS240
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	HPE Integrity MC990 X
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v2	HPE ProLiant BL460c Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	HPE ProLiant BL460c Gen9
Oracle Linux 7.6 64-bit	AMD Opteron 6300-series	HPE ProLiant BL465c Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v2	HPE ProLiant BL660c Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v3	HPE ProLiant BL660c Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL120 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL160 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL180 Gen9
Oracle Linux 7.6 64-bit	Intel ® Pentium® G2120 & Intel® Xeon® E3-1200 v2	HPE ProLiant DL320e Gen8
Oracle Linux 7.6 64-bit	Intel® Pentium® G3200-series/G3420, Core i3-4100-series/Intel® Xeon® E3-12 v3	HPE ProLiant DL320e Gen8 v2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL360 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100/4100/3100 Processors	HPE ProLiant DL360 Gen10

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2400/E5-2400 v2	HPE ProLiant DL360e Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL360p Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant DL380 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2400/E5-2400 v2	HPE ProLiant DL380e Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600/E5-2600 v2	HPE ProLiant DL380p Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100/4100/3100 Processors	HPE ProLiant DL380 Gen10
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600/E5-4600 v2	HPE ProLiant DL560 Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon®E5-4600 v3 & v4	HPE ProLiant DL560 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8170 Processors	HPE ProLiant DL560 Gen10
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2/E7-8800 v2	HPE ProLiant DL580 Gen8
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3/E7-8800 v3	HPE ProLiant DL580 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® X7560, X6550, E6540, E7520	HPE ProLiant DL980 G7
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	HPE ProLiant ML350 Gen9
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	HPE ProLiant XL450 Gen9 (Apollo 4500)
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	HPE Synergy 480 Gen9 Compute Module
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100/4100/3100 Processors	HPE Synergy 480 Gen10 Compute Module
Oracle Linux 7.6 64-bit	Intel® Xeon®E7-4800 v4/E7-8800 v4	HPE Synergy 620 Gen9 Compute Module
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100 Processors	HPE Synergy 660 Gen10 Compute Module
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	HPE Synergy 680 Gen9 Compute Module
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer 1288H V5
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer 2288H V5
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer CH121 V5
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer CH121L V5
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer CH242 V5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Huawei FusionServer RH2288H V3
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Huawei FusionServer XH321 V5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5170M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Inspur Yingxin NF5180M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Inspur Yingxin NF5180M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5240M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v2	Inspur Yingxin NF5270M3
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5270M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5280M4
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable Processors	Inspur Yingxin NF5280M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5460M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3 & v4/E7-8800 v3 & v4	Inspur Yingxin NX8480M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Lenovo System x3650 M5
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	Lenovo System x3850 X6

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100/4100/3100 Processors	Lenovo ThinkSystem SD530
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100/4100/3100 Processors	Lenovo ThinkSystem SN550
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100 Processors	Lenovo ThinkSystem SN850
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100 Processors	Lenovo ThinkSystem SR850
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100 Processors	Lenovo ThinkSystem SR860
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/5100 Processors	Lenovo ThinkSystem SR950
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC Express 5800/A1040d
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC Express 5800/A2010d
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC Express 5800/A2020d
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC Express 5800/A2040d
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4800 v4/E7-8800 v4	NEC Express 5800/R120g-1M
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	NEC Express 5800/R120g-2M
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC NX7700x/A4010M-4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC NX7700x/A4012L-1
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800/4800 v4	NEC NX7700x/A4012L-1D
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/E7-8800 v4	NEC NX7700x/A4012L-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800/4800 v4	NEC NX7700x/A4012L-2D
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3/E7-8800 v3	NEC NX7700x/A4012M-4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Oracle Netra Server X5-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Oracle Server X5-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Oracle Server X5-2L
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Oracle Server X5-4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3	Oracle ServerX5-8
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Oracle Server X6-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Oracle Server X6-2L
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v4	Oracle Server X6-2M
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/4100 Processors	Oracle Server X7-2
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100/4100 Processors	Oracle Server X7-2L
Oracle Linux 7.6 64-bit	Intel® Xeon® Scalable 8100/6100 Processors	Oracle Server X7-8
Oracle Linux 7.6 64-bit	Intel® Xeon® x7500-series	Oracle Sun Fire X4470
Oracle Linux 7.6 64-bit	Intel® Xeon® x7500-series	Oracle Sun Fire X4800
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800	Oracle Sun Server X2-8
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800	Oracle Sun Server X2-4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600	Oracle Sun Server X3-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600	Oracle Sun Server X3-2L
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v2	Oracle Sun Server X4-2
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v2	Oracle Sun Server X4-2L
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Oracle Sfun Server X4-4
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v2	Oracle Sun Server X4-8
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-8800 v3 & v4	SGI UV 300RL
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v4/ E7-8800 v3 & v4	SGI UV 300
Oracle Linux 7.6 64-bit	AMD Opteron™ 6000	Sugon A840-G10
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Sugon CB50-G20
Oracle Linux 7.6 64-bit	AMD Opteron™ 6000	Sugon CB85-G10

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Sugon CB85-G10
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2	Sugon CB80-G20
Oracle Linux 7.6 64-bit	Intel Xeon E7-8800/4800-v3 Series	Sugon CB80-G25
Oracle Linux 7.6 64-bit	AMD Opteron™ 6300	Sugon CB85-G10
Oracle Linux 7.6 64-bit	Intel® Xeon® 6100, 5100, 4100, 3100	Sugon I420-G30
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Sugon I610-G20
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Sugon I620-G20
Oracle Linux 7.6 64-bit	Intel® Xeon® 8100	Sugon I620-G30
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v3 & v4	Sugon I840-G20
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2	Sugon I840-G25
Oracle Linux 7.6 64-bit	Intel® Xeon® E7-4800 v2 & v3/E7-8800 v2 & v3	Sugon I980-G20
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Sugon TC4600T
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Supermicro SuperServer SYS-6018U-TR4T+

Table 7: Vendor Affirmed Operating Environment

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.

6.3 Operational Environment Policy

The operating system is restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded).

The application that makes calls to the Modules is the single user of the Modules, even when the application is serving multiple clients.

During module operation, the ptrace(2) system call, the debugger (gdb(1)), and strace(1) shall not be used. In addition, other tracing mechanisms offered by the Linux environment, such as Dtrace, ftrace or systemtap, shall not be used.

7. Roles, Services and Authentication

7.1 Roles

The Oracle Linux OpenSSL Cryptographic Module supports 2 roles as required by FIPS PUB 140-2. These roles are a Crypto Officer Role and a User Role. Both roles are implicitly assumed by the entity accessing services implemented by the Module. The module does not support authentication mechanisms.

7.2 FIPS Approved Operator Services and Descriptions

The below table provides a full description of FIPS Approved services provided by the module and lists the roles allowed to invoke each service. In the table below, the "U" represents a User Role, and "CO" denotes a Crypto Officer role.

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X		Symmetric Encryption/Decryption	Encrypts or decrypts a block of data using 3-Key Triple-DES or AES	AES or 3-Key Triple-DES Key	R, W, X
X		Asymmetric Key Generation	Generate RSA, DSA/ Diffie-Hellman, ECDSA/ EC Diffie-Hellman asymmetric keys	RSA, DSA/ Diffie-Hellman, ECDSA/EC Diffie-Hellman keys	R, W, X
X		Digital Signature Generation and Verification	Sign and verify operations	RSA, DSA, and ECDSA keys	R, W, X
X		TLS Network Protocol	Provide data encryption and authentication over TLS network protocol	AES, Triple-DES, and HMAC keys.	R, W, X
X		TLS Key Agreement	Negotiate a TLS key agreement secure channel	AES or Triple-DES key, RSA, DSA or ECDSA private key, HMAC Key, shared secrete, Diffie-Hellman and EC Diffie-Hellman Private keys	R, W, X
X		Asymmetric Encryption/Decryption	Encrypts or decrypts using Approved RSA key size	RSA key pair	R, W, X
X		Certificate Management Handling	Management of key properties within certificates.	RSA, DSA, and ECDSA private keys	R, W, X
X		Keyed Hash (HMAC)	Sign and or authenticate data using HMAC-SHA	HMAC Key	R, W, X
X		Keyed Hash (CMAC)	Encrypt and sign data using CMAC.	AES or 3-Key Triple-DES Key	R, W, X
X		Hash (SHS)	Hash a block of data.	None	R, W, X
X		Random Number Generation	Generate random numbers based on the NIST SP 800-90A Standard	Entropy input string and internal State	R, W, X
X		Show Status	Show status of the module state	None	X

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X		Self-Test	Initiate power-on self-tests	None	R, X
X		Zeroize	Zeroize all critical security parameters	All keys and CSP's	Z
	X	Module Installation	Installation of the module	None	R, W

R – Read, W – Write, X – Execute, Z - Zeroize

Table 8: FIPS Approved Services and Descriptions

7.3 Non-FIPS Approved Services and Descriptions

The following table lists the non-Approved services available in non-FIPS mode. Security services listed in the table below, make use of non-approved cryptographic algorithms. Use of any of these services in Table 9 will put the module in the non-Approved mode implicitly. The algorithms associated with these services are specified in section 3.5.

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X		Asymmetric Encryption/Decryption	Encrypts or decrypts using non-Approved RSA key size	RSA key pair	R, W, X
X		Symmetric Encryption/Decryption	Encrypts or decrypts using non-Approved algorithms	AES XTS, Camellia, CAST, DES, IDEA, RC2, RC4, RC5 keys	R, W, X
X		Digital Signature Generation and Verification	Sign or verify operations with non-Approved RSA or DSA key lengths	RSA key < 2048 DSA keys not listed in Table 2 Signature Generation with SHA-1	R, W, X
X		TLS Key Agreement	Negotiate a TLS key agreement secure channel with non-Approved key sizes	RSA/ Diffie-Hellman key < 2048 EC Diffie-Hellman with P-192	R, W, X
X		Asymmetric Key Generation	Generation of non-Approved RSA and DSA keys	RSA key < 2048 DSA keys not listed in Table 2	R, W, X
X		Random Number Generation	Generation of random numbers using the ANSI X9.31 PRNG	seed, seed key	R, W, X
X		Hash	Hashing using non-Approved hash functions that include MD2, MD4, MD5, MDC2, RIPEMD, Whirlpool	None	R, W, X
X		J-PAKE Key Agreement	Password authenticated key agreement using J-PAKE	J-PAKE key pair	R, W, X

Table 9: Non-FIPS Approved Services and Descriptions

7.4 Operator Authentication

The module does not support operator authentication mechanisms.

8. Key and CSP Management

The following keys, cryptographic key components and other critical security parameters are contained in the module.

CSP Name	Generation/Input	Use
AES Key (128, 192, 256 bits)	The Key is passed into the module via API input parameter	Encrypt/Decrypt operations Used to generate and verify MAC's with AES as part of the CMAC algorithm.
Triple-DES Keys (192 bits)		Used for Encrypt/Decrypt operations. Used for generating and verifying MAC's with Triple-DES as part of the CMAC algorithm.
HMAC Key (≥ 112 bits)		HMAC keys used to generate and verify MAC's on data.
RSA Key pair (2048, 3072, 4096 bits)	Keys are generated using FIPS 186-4 and the random value used in the key generation is generated using SP800- 90A DRBG	RSA public/private keys used to sign and verify data. RSA private key used for key decryption as part of key wrapping.
DSA/ Diffie-Hellman Key pair (2048 , 3072, 4096 bits)		DSA public/private keys used to sign and verify data. Diffie-Hellman key pair used as part of the key agreement protocol.
ECDSA/ EC Diffie-Hellman Key pair (P-224,P-256, P-384, P-521)		ECDSA public/private keys used to sign and verify data. EC Diffie-Hellman key pair used as part of the key agreement protocol.
Shared secret	Generated during the Diffie-Hellman or EC Diffie-Hellman key agreement.	Used to derive the required keys by applying key derivation function.
Entropy input string	Obtained from NDRNG	Entropy input strings used as part of the DRBG process.
DRBG Internal stated (V,C , Key value)	During DRBG initialization.	V and key are used as part of HMAC and CTR DRBG process. V and C are used as part of HASH DRBG process.
TLS Pre-Master Secret and Master Secret	Established during the TLS handshake	Used as part of the TLS key establishment protocol

Table 10: CSP Table

8.1 Random Number Generation

The Module provides an SP800-90A-compliant DRBG for the creation of key components of asymmetric keys, and random number generation.

The Module uses NDRNG from /dev/urandom as a source of entropy for seeding the DRBG. The NDRNG is provided by the operational environment (i.e., Linux RNG), which is within the module's physical boundary but outside of the module's logical boundary. The NDRNG provides at least 130 bits of entropy to the DRBG. *The module generates cryptographic keys whose strengths are modified by available entropy.*

The module performs Health Testing on the SP800-90A random bit generator encompassing instantiation, generation, and reseeding functions. The underlying operating system performs the continuous test on the NDRNG.

8.2 Key Generation

For generating RSA, DSA/DH and ECDSA/ECDH keys, the module implements asymmetric key generation services compliant with [FIPS186-4], and using DRBG compliant with [SP800-90A]. A seed (i.e. the random value) used in asymmetric key generation is obtained from [SP800-90A] DRBG. The module does not implement symmetric key generation. In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per NIST SP 800-133 (vendor affirmed). The resulting symmetric key or asymmetric seed is an unmodified output from a DRBG.

The module does not support manual key entry or intermediate key generation output. In addition, the module does not produce key output outside its physical boundary. The keys can be entered or output from the module in plaintext form via API parameters, to and from the calling application only.

8.3 Key/CSP Storage

Public and private keys are provided to the Module by the calling process, and are destroyed when released by the appropriate API function calls. The Module does not perform persistent storage of keys.

8.4 Key/CSP Zeroization

The application is responsible for calling the appropriate destruction functions from the OpenSSL API. The destruction functions then overwrite the memory occupied by keys with zeros and deallocates the memory with the free() call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process.

8.5 Key Establishment

The module provides Diffie-Hellman and EC Diffie-Hellman key agreement schemes. These key agreement schemes are also used as part of the TLS protocol key exchange. The module also provides AES and RSA key wrapping. RSA key wrapping is also used as part of the TLS protocol key exchange.

The AES key wrapping provides between 128 and 256 bits of encryption strength. The security strength for RSA, Diffie-Hellman and EC Diffie-Hellman key agreement schemes is listed in Table 3.

9. Self-Tests

FIPS 140-2 requires that the Module performs self-tests to ensure the integrity of the Module, and the correctness of the cryptographic functionality at start up. In addition, conditional tests are required during operational stage of the module. All of these tests are listed and described in this section. See section 10.3 for descriptions of possible self-test errors and recovery procedure.

9.1 Power-Up Self-Tests

The Module performs power-up self-tests automatically during loading of the module by making use of default entry point (DEP) and no operator intervention is required. The module is not available for use until successful completion of power-up self-tests. Hence input, output, or any cryptographic functions cannot be performed while the Module is executing self-tests. The integrity of the module binary is verified using a HMAC SHA-256. The HMAC value is computed at build time and stored in the hmac file. The value is recalculated at runtime and compared against the stored value. If the comparison succeeds, then the remaining power-up self-test (consisting of the algorithm-specific Known Answer Tests) are performed. On successful completion of the power-up tests, the module becomes operational and crypto services are available. If any of the tests fails module transitions to error state and subsequent calls to the Module will fail - thus no further cryptographic operations will be possible.

Algorithm	Test
AES	KAT, encryption and decryption are tested separately.
Triple-DES	KAT, encryption and decryption are tested separately.
DSA	PCT, sign and verify.
RSA	KAT, signature generation and verification are tested separately.
ECDSA	PCT, sign and verify
Diffie-Hellman	Primitive "Z" Computation KAT
EC Diffie-Hellman	Primitive "Z" Computation KAT
SP 800-90A CTR_DRBG	KAT
SP 800-90A Hash_DRBG	KAT
SP 800-90A HMAC_DRBG	KAT
HMAC	(SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) KAT
SHA	(1, 256, 512) KAT
CMAC	KAT
Module Integrity	HMAC-SHA-256

Table 11: Power-On Self-Tests

9.2 Conditional Self-Tests

Conditional tests are performed during operational state of the module when the respective crypto functions are used. If any of the conditional tests fails, module transitions to error state.

Algorithm	Test
DSA Key generation	Pairwise Consistency Test: signature generation and verification
ECDSA Key generation	Pairwise Consistency Test: signature generation and verification
RSA Key generation	Pairwise Consistency Test: signature generation and verification, encryption and decryption
DRBG NIST SP 800-90A	SP800-90A Health Tests

Table 12: Conditional Self-Tests

9.3 On-Demand self-tests

The module provides the Self-Test service to perform self-tests on demand. On demand self-tests can be invoked by powering-off and reloading the module. This service performs the same cryptographic algorithm tests executed during power-up. During the execution of the on-demand self-tests, crypto services are not available and no data output or input is possible

10. Crypto-Officer and User Guidance

This section provides guidance for the Cryptographic Officer and the User to maintain proper use of the module per FIPS 140-2 requirements.

10.1 Crypto-Officer Guidance

The version of the RPM containing the validated module is stated in section 3.1 above. The RPM package of the Module can be installed by standard tools recommended for the installation of Oracle packages on an Oracle Linux system (for example, yum, RPM, and the RHN remote management tool). The integrity of the RPM is automatically verified during the installation of the Module and the Crypto Officer shall not install the RPM file if the Oracle Linux Yum Server indicates an integrity error. The RPM files listed in section 3 are signed by Oracle and during installation; Yum performs signature verification which ensures as secure delivery of the cryptographic module. If the RPM packages are downloaded manually, then the CO should run 'rpm –K <rpm-file-name>' command after importing the builder's GPG key to verify the package signature. In addition, the CO can also verify the hash of the RPM package to confirm a proper download.

The OpenSSL static libraries libcrypto.a and libssl.a in openssl-static package are not approved to be used. The applications must be dynamically linked to run the OpenSSL.

To configure the operating environment to support FIPS Approved mode, perform the following steps:

1. Insure that the system is registered with the unbreakable Linux Network (ULN) and that the OL7_X86_64_latest channel is enabled

   ```
   # yum-config-manager –enable ol7_latest
   ```

2. Install the dracut-fips package:

   ```
   # yum install dracut-fips
   ```

3. Install the dracut-fips-aesni package (if AES-NI is supported):

   To check if AES-NI is supported run:

   ```
   # grep aes /proc/cpuinfo
   ```

   If it is supported, run:

   ```
   # yum install dracut-fips-aesni
   ```

4. Recreate the INITRAMFS image:

   ```
   # dracut -f
   ```

5. Perform the following steps to configure the boot loader so that the system boots into FIPS mode:

   a) Identify the boot partition and the UUID of the partition. If /boot or /boot/efi resides on a separate partition, the kernel parameter boot=<partition of /boot or /boot/efi> must be supplied. The partition can be identified with the command:

   ```
   # df /boot or df /boot/efi
   ```

```
Filesystem   1K-blocks   Used    Available   Use%   Mounted on
/dev/sda1    233191      30454   190296      14%    /boot
```

```
# blkid /dev/sda1
```

```
/dev/sda1: UUID="6046308a-75fc-418e-b284-72d8bfad34ba" TYPE="xfs"
```

   b) As the root user, edit the /etc/default/grub file as follows:

i. Add the fips=1 option to the boot loader configuration.

```
 GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
 rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
vconsole.keymap=uk rhgb quiet fips=1"
```

ii. If the contents of /boot reside on a different partition to the root partition, you must use the boot=UUID=boot_UUID line to the boot loader configuration to specify the device that should be mounted onto /boot when the kernel loads.

```
GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
    rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
    vconsole.keymap=uk rhgb quiet
    boot=UUID=6046308a-75fc-418e-b284-72d8bfad34ba fips=1"
```

iii. Save the changes.

This is required for FIPS to perform kernel validation checks, where it verifies the kernel against the provided HMAC file in the /boot directory.

Note:
On systems that are configured to boot with UEFI, /boot/efi is located on a dedicated partition as this is formatted specifically to meet UEFI requirements. This does not automatically mean that /boot is located on a dedicated partition.

Only use the boot= parameter if /boot is located on a dedicated partition. If the parameter is specified incorrectly or points to a non-existent device, the system may not boot.

If the system is no longer able to boot, you can try to modify the kernel boot options in grub to specify an alternate device for the boot=UUID=boot_UUID parameter, or remove the parameter entirely.

6. Rebuild the GRUB configuration as follows:

On BIOS-based systems, run the following command:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

On UEFI-based systems, run the following command:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

To ensure proper operation of the in-module integrity verification, prelinking must be disabled on all system files. By default, the prelink package is not installed on the system. However, if it is installed, disable prelinking on all libraries and binaries as follows:

Set PRELINKING=no in the /etc/sysconfig/prelink configuration file.

If the libraries were already prelinked, undo the prelink on all of the system files as follows:

```
# prelink –u –a
```

7. Reboot the system

8. Verify that FIPS Mode is enabled by running the command:

   ```
   # cat /proc/sys/crypto/fips_enabled
   ```

   The response should be “1”

The version of the RPM containing the validated Modules is the version listed in Section 3. The integrity of the RPM is automatically verified during the installation of the Modules and the Crypto Officer shall not install the RPM file if the RPM tool indicates an integrity error.

10.2 User Guidance

In order to run the module in FIPS mode, only the FIPS approved or allowed services listed in table 8 or the validated or allowed cryptographic algorithms/security functions listed in Table 2 and 3 should be used.

Interpretation of the return code is the responsibility of the host application.

ENGINE_register_*, ENGINE_set_default_* and FIPS_mode_set(0) function calls are prohibited.

10.2.1 TLS and Diffie-Hellman

The TLS protocol implementation provides both, the server and the client sides. As required by SP800-131Ar1, Diffie-Helllman with keys smaller than 2048 bits must not be used. The TLS protocol lacks the support to negotiate the used Diffie-Hellman key sizes. To ensure full support for all TLS protocol versions, the TLS client implementation of the cryptographic Module accepts Diffie-Hellman key sizes smaller than 2048 bits offered by the TLS server. The TLS server implementation of the cryptographic Module allows the application to set the Diffie-Hellman key size. The server side must always set the DH parameters with the API call of

```
SSL_CTX_set_tmp_dh(ctx, dh)
```

For complying with the requirement to not allow Diffie-Hellman key sizes smaller than 2048 bits, the Crypto Officer must ensure that:

- in case the Module are used as TLS server, the Diffie-Hellman parameters (dh argument) of the aforementioned API call must be 2048 bits or larger;
- in case the Module are used as TLS client, the TLS server must be configured to only offer Diffie-Hellman keys of 2048 bits or larger.

10.2.2 Random Number Generator

The OpenSSL API call of RAND_cleanup must not be used. This call will clean up the internal DRBG state. This call also replaces the DRBG instance with the non-FIPS Approved SSLeay Deterministic Random Number Generator when using the RAND_* API calls.

10.2.3 AES GCM IV

The IV generation method is user selectable and may be computed in the following ways:

1. Conforming to IG A.5, scenario #1 (for TLS 1.2): Comply with the provision of a peer-to-peer industry standard. Specifically, following RFC 5288 for TLS. The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key in accordance with RFC 5246.
2. Conforming to IG A.5, scenario #3 (for non-TLS 1.2): The IV is constructed in its entirety internally deterministically, consisting of 96 bits as specified in SP800-38D, section 8.2.1.

It is the responsibility of the user to determine which method to use.

In case the Modules' power is lost and then restored, the key used for the AES GCM encryption/decryption shall be re-distributed.

10.2.4 AES-XTS Guidance

The length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks that is 16MB of data. AES-XTS mode shall be used for storage purposes only. The calling process is responsible to ensure Key_1 is not equal to Key_2 when using AES-XTS.

10.2.5 Triple-DES Keys

According to IG A.13, the same Triple-DES key shall not be used to encrypt more than 2^{20} 64- bit blocks of data.

10.2.6 RSA and DSA Keys

The Modules allow the use of 1024 bit RSA and DSA keys for legacy purposes, including signature generation.

Application can enforce the key generation bit length restriction for RSA and DSA keys by setting the environment variable OPENSSL_ENFORCE_MODULUS_BITS. This environment variable ensures that 1024 bit keys cannot be generated.

10.3 Handling Self-Test Errors

The Module transition to error state when any of self-tests or conditional tests fails. The application must be restarted to recover from these errors. Following are the error messages specific to self-test failure:

FIPS_R_FINGERPRINT_DOES_NOT_MATCH – The integrity verification check failed
FIPS_R_SELFTEST_FAILED – a known answer test failed
FIPS_R_TEST_FAILURE – a known answer test failed (RSA); pairwise consistency test failed (DSA)
FIPS_R_PAIRWISE_TEST_FAILED – a pairwise consistency test failed during EC/DSA or RSA key generation
FIPS_R_SELFTEST_FAILURE – the DRBG Health Test failed

These errors are reported through the regular ERR interface of the Module and can be queried by functions such as ERR_get_error(). See the OpenSSL manual page for the function description.

When the Module is in error state, output is inhibited and no crypto operations are available. Any calls to the crypto functions in error state will return error message: 'FATAL FIPS SELFTEST FAILURE' on stderr and the application is terminated with the abort() call.

The only way to recover from the error state is to reload the module and restart the application. If failures persist, the Module must be reinstalled. If downloading the software, make sure to verify the package hash to confirm a proper download.

11. Mitigation of Other Attacks

RSA is vulnerable to timing attacks. In a setup where attackers can measure the time of RSA decryption or signature operations, blinding must be used to protect the RSA operation from that attack. The API function of RSA_blinding_on turns blinding on for key rsa and generates a random blinding factor. The random number generator must be seeded prior to calling RSA_blinding_on.

Weak Triple-DES keys are detected as follows:

```
/* Weak and semi week keys as taken from
* %A D.W. Davies
* %A W.L. Price
* %T Security for Computer Networks
* %I John Wiley & Sons
* %D 1984
* Many thanks to smb@ulysses.att.com (Steven Bellovin) for the reference
* (and actual cblock values).
*/
#define NUM_WEAK_KEY 16
static const DES_cblock weak_keys[NUM_WEAK_KEY]={
/* weak keys */
            {0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01},
            {0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE},
            {0x1F,0x1F,0x1F,0x1F,0x0E,0x0E,0x0E,0x0E},
            {0xE0,0xE0,0xE0,0xE0,0xF1,0xF1,0xF1,0xF1},
            /* semi-weak keys */
            {0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE},
            {0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01},
            {0x1F,0xE0,0x1F,0xE0,0x0E,0xF1,0x0E,0xF1},
            {0xE0,0x1F,0xE0,0x1F,0xF1,0x0E,0xF1,0x0E},
            {0x01,0xE0,0x01,0xE0,0x01,0xF1,0x01,0xF1},
            {0xE0,0x01,0xE0,0x01,0xF1,0x01,0xF1,0x01},
            {0x1F,0xFE,0x1F,0xFE,0x0E,0xFE,0x0E,0xFE},
            {0xFE,0x1F,0xFE,0x1F,0xFE,0x0E,0xFE,0x0E},
            {0x01,0x1F,0x01,0x1F,0x01,0x0E,0x01,0x0E},
            {0x1F,0x01,0x1F,0x01,0x0E,0x01,0x0E,0x01},
            {0xE0,0xFE,0xE0,0xFE,0xF1,0xFE,0xF1,0xFE},
            {0xFE,0xE0,0xFE,0xE0,0xFE,0xF1,0xFE,0xF1}};
```

Please note that there is no weak key detection by default. The caller can explicitly set the DES_check_key to 1 or call DES_check_key_parity() and/or DES_is_weak_key() functions on its own.

Acronyms, Terms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
AVX	Advanced Vector Extensions
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	Error Detection Code
GPG	Gnu Privacy Guard
HMAC	(Keyed) Hash Message Authentication Code
IKE	Internet Key Exchange
IPSEC	Internet Protocol Security
KAT	Known Answer Test
KDF	Key Derivation Function
NDRNG	Non-Deterministic Random Number generator
NIST	National Institute of Standards and Technology
PAA	Processor Algorithm Acceleration
POST	Power On Self Test
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
PUB	Publication
SHA	Secure Hash Algorithm
SSSE3	Supplemental Streaming SIMD Extensions 3
UEK	Oracle Linux Unbreakable Enterprise Kernel
VPAES	AES with Vector Permutations
TLS	Transport Layer Security

Table 13: Acronyms

References

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html. More information describing the module can be found on the Oracle web site at https://www.oracle.com/linux/ .

This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is "Oracle - Proprietary" and is releasable only under appropriate non-disclosure agreements.

Document	Author	Title
FIPS PUB 140-2	NIST	FIPS PUB 140-2: Security Requirements for Cryptographic Modules
FIPS IG	NIST	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program
FIPS PUB 140-2 Annex A	NIST	FIPS 140-2 Annex A: Approved Security Functions
FIPS PUB 140-2 Annex B	NIST	FIPS 140-2 Annex B: Approved Protection Profiles
FIPS PUB 140-2 Annex C	NIST	FIPS 140-2 Annex C: Approved Random Number Generators
FIPS PUB 140-2 Annex D	NIST	FIPS 140-2 Annex D: Approved Key Establishment Techniques
DTR for FIPS PUB 140-2	NIST	Derived Test Requirements (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules
NIST SP 800-67	NIST	Recommendation for the Triple Data Encryption Algorithm TDEA Block Cipher
FIPS PUB 197	NIST	Advanced Encryption Standard
FIPS PUB 198-1	NIST	The Keyed Hash Message Authentication Code (HMAC)
FIPS PUB 186-4	NIST	Digital Signature Standard (DSS)
FIPS PUB 180-4	NIST	Secure Hash Standard (SHS)
NIST SP 800-131Ar1	NIST	Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes
PKCS#1	RSA Laboratories	PKCS#1 v2.1: RSA Cryptographic Standard
RFC 5288	https://tools.ietf.org/html/rfc5288	AES Galois Counter Mode (GCM) Cipher Suites for TLS

Table 14: References