

**ORACLE®**

Linux

FIPS 140-2 Non-Proprietary Security Policy

---

## Oracle Linux 7 NSS Cryptographic Module

FIPS 140-2 Level 1 Validation

Software Version: R7-4.0.0

Date: May 5, 2023



**Title:** Oracle Linux 7 NSS Cryptographic Module Security Policy

**Date:** May 5, 2023

**Author:** Oracle Security Evaluations – Global Product Security

**Contributing Authors:**

Oracle Linux Engineering

Oracle Corporation

World Headquarters

2300 Oracle Way

Austin, TX 78741

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

[www.oracle.com](http://www.oracle.com)



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2023, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. Oracle specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may be reproduced or distributed whole and intact including this copyright notice.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

**Hardware and Software, Engineered to Work Together**



## TABLE OF CONTENTS

| Section   | Title   | Page      |
|---|---|-----------|
| <b>1. Introduction</b>                            |   | <b>1</b>  |
| 1.1   | Overview  | 1         |
| 1.2   | Document Organization                                     | 1         |
| <b>2. Oracle Linux 7 NSS Cryptographic Module</b> |   | <b>2</b>  |
| 2.1   | Functional Overview                                       | 2         |
| 2.2   | FIPS 140-2 Validation Scope                               | 2         |
| <b>3. Cryptographic Module Specification</b>      |   | <b>3</b>  |
| 3.1   | Definition of the Cryptographic Module                    | 3         |
| 3.2   | Definition of the Physical Cryptographic Boundary         | 4         |
| 3.3   | Approved Security Functions                               | 4         |
| 3.4   | Non-Approved but Allowed Security Functions               | 6         |
| 3.5   | Non-Approved Security Functions                           | 6         |
| <b>4. Module Ports and Interfaces</b>             |   | <b>8</b>  |
| 4.1   | PKCS#11   | 8         |
| 4.2   | Inhibition of Data Output                                 | 8         |
| 4.3   | Disconnecting the Output Data Path from the Key Processes | 9         |
| <b>5. Physical Security</b>                       |   | <b>10</b> |
| <b>6. Operational Environment</b>                 |   | <b>11</b> |
| 6.1   | Tested Environments                                       | 11        |
| 6.2   | Vendor Affirmed Environments                              | 11        |
| 6.3   | Operational Environment Policy                            | 15        |
| <b>7. Roles, Services and Authentication</b>      |   | <b>17</b> |
| 7.1   | Roles   | 17        |
| 7.2   | FIPS Approved Operator Services and Descriptions          | 17        |
| 7.3   | Non-FIPS Approved Services and Descriptions               | 22        |
| 7.4   | Operator Authentication                                   | 23        |
| 7.4.1   | Role Assumption   | 23        |
| 7.4.2   | Strength of Authentication Mechanism                      | 24        |
| <b>8. Key and CSP Management</b>                  |   | <b>25</b> |
| 8.1   | Random Number Generation                                  | 25        |
| 8.2   | Key/CSP Storage   | 26        |
| 8.3   | Key/CSP Zeroization                                       | 26        |
| <b>9. Self-Tests</b>                              |   | <b>27</b> |
| 9.1   | Power-Up Self-Tests                                       | 27        |
| 9.2   | Conditional Self-Tests                                    | 27        |
| <b>10. Crypto-Officer and User Guidance</b>       |   | <b>28</b> |
| 10.1  | Crypto-Officer Guidance                                   | 28        |
| 10.1.1  | Access to Audit Data                                      | 30        |
| 10.2  | User Guidance   | 30        |



|            |  |           |
|------------|--|-----------|
| 10.2.1     | TLS Operations.....                            | 31        |
| 10.2.2     | RSA and DSA Keys.....                          | 31        |
| 10.2.3     | Triple-DES keys .....                          | 32        |
| 10.3       | Handling Self-Test Errors .....                | 32        |
| <b>11.</b> | <b>Mitigation of Other Attacks.....</b>        | <b>33</b> |
|            | <b>Acronyms, Terms and Abbreviations .....</b> | <b>34</b> |
|            | <b>References .....</b>                        | <b>35</b> |

## List of Tables

|  |    |
|--|----|
| Table 1: FIPS 140-2 Security Requirements.....                     | 2  |
| Table 2: FIPS Approved Security Functions.....                     | 6  |
| Table 3: Non-Approved but Allowed Security Functions .....         | 6  |
| Table 4: Non-Approved and Disallowed Functions .....               | 7  |
| Table 5: Mapping of FIPS 140-2 Logical Interfaces .....            | 8  |
| Table 6: Tested Operating Environment.....                         | 11 |
| Table 7: Vendor Affirmed Operational Environments .....            | 15 |
| Table 8: FIPS Approved Operator Services and Descriptions .....    | 21 |
| Table 9: Non-FIPS Approved Operator Services and Descriptions..... | 23 |
| Table 10: CSP Table .....  | 25 |
| Table 11: Power-On Self-Tests .....                                | 27 |
| Table 12: Conditional Self-Tests.....                              | 27 |
| Table 13: Mitigation of Other Attacks .....                        | 33 |
| Table 14: Acronyms.....  | 34 |

## List of Figures

|   |   |
|---|---|
| Figure 1: Oracle Linux 7 NSS Logical Cryptographic Boundary ..... | 3 |
| Figure 2: Oracle Linux 7 NSS Hardware Block Diagram.....          | 4 |

## 1. Introduction

### 1.1 Overview

This document is the Security Policy for the Oracle Linux 7 NSS Cryptographic Module by Oracle Corporation. This Security Policy specifies the security rules under which the module shall operate to meet the requirements of FIPS 140-2 Level 1. It also describes how the Oracle Linux 7 NSS Cryptographic Module functions in order to meet the FIPS 140-2 requirements, and the actions that operators must take to maintain the security of the module.

This Security Policy describes the features and design of the Oracle Linux 7 NSS 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/CCCS 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 FIPS 140-2 Submission Package contains:

- Oracle Linux 7 NSS 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 7 NSS Cryptographic Module

### 2.1 Functional Overview

The Oracle Linux 7 NSS Cryptographic Module (hereafter referred to as the “module”) is a set of libraries designed to support cross-platform development of security-enabled applications. Applications built with the Oracle Linux 7 NSS Cryptographic Module can support 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. It combines a vertical stack of Oracle Linux components intended to limit the external interface each separate component may provide. The Oracle Linux 7 NSS Cryptographic Module is distributed with the Oracle Linux open-source distributions. The module provides a C-language Application Program Interface (API) for use by other processes that require cryptographic functionality.

Oracle Linux 7 NSS Cryptographic Module supports 2 types of cryptographic implementations:

- a) NSS in Native C Programming Language; and
- b) AES-NI for X86 processors.

### 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     | 2     |
| 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 7 NSS Cryptographic Module with version R7-4.0.0 is defined as a software only 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 signature files, which are delivered through the Package Manager (RPM) as listed below:

- nss-softokn RPM file with version [nss-softokn-3.36.0-5.0.1.el7\\_5.x86\\_64](#) which contains the following files:
  - /usr/lib64/libnssdbm3.chk (64 bits)
  - /usr/lib64/libnssdbm3.so (64 bits)
  - /usr/lib64/libsoftokn3.chk (64 bits)
  - /usr/lib64/libsoftokn3.so (64 bits)
  
- nss-softokn-freebl RPM file with version [nss-softokn-freebl-3.36.0-5.0.1.el7\\_5.x86\\_64](#), which contains the following files:
  - /lib64/libfreeblpriv3.chk (64 bits)
  - /lib64/libfreeblpriv3.so (64 bits)

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

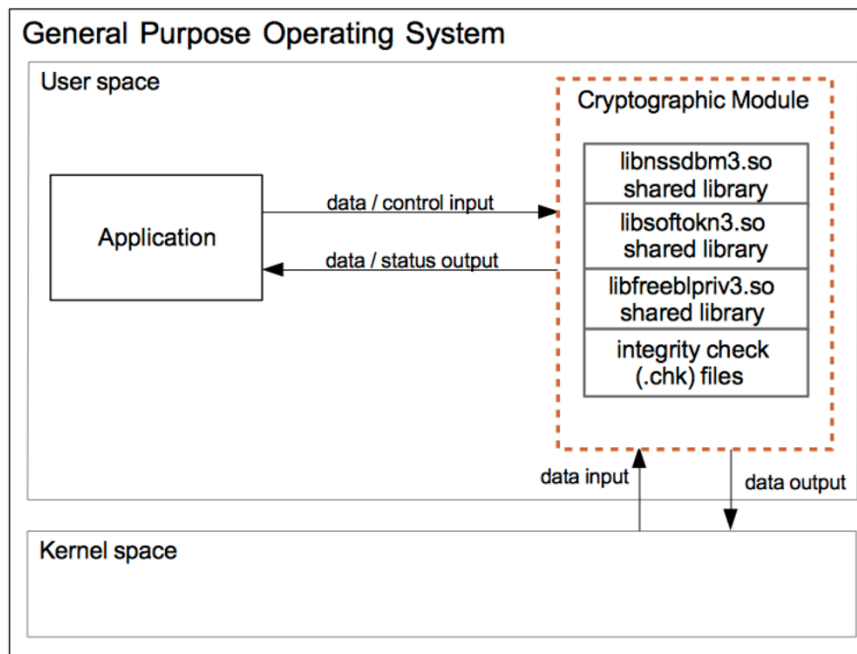
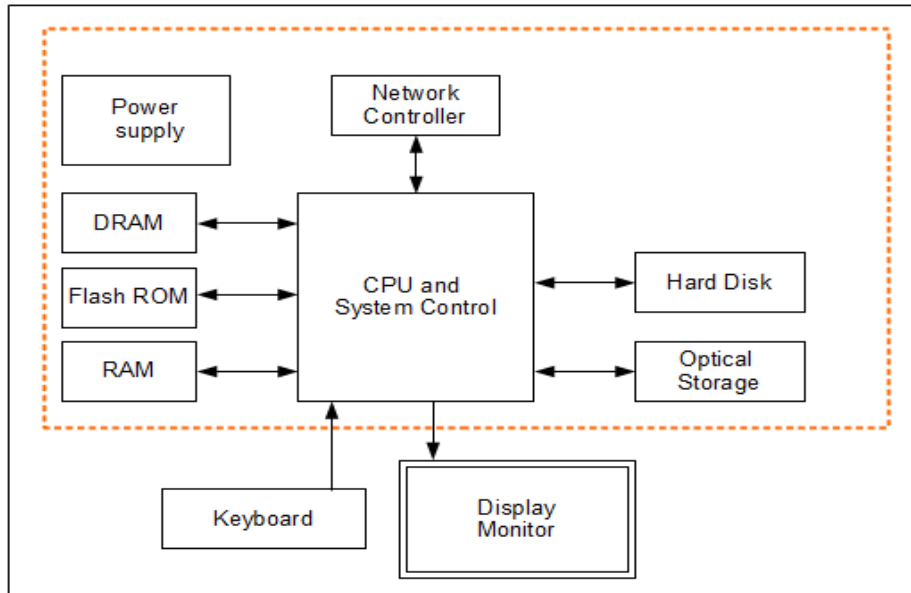


Figure 1: Oracle Linux 7 NSS Logical Cryptographic Boundary



### 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 7 NSS Hardware Block Diagram**

### 3.3 Approved Security Functions

The module supports two modes of operation: FIPS Approved mode and non-Approved mode.

When the module is powered on, the power-up self-tests are executed automatically without any operator intervention. If the power-up self-tests complete successfully, the module will be in FIPS Approved mode as long as only approved services are called. In Approved mode, only Approved algorithms (as listed in Table 2) and non-approved but Allowed algorithms (as listed in Table 3) can be used.

| Approved Security Functions |   | Certificate          |
|-----------------------------|---|----------------------|
| <b>Symmetric Algorithms</b> |   |                      |
| AES                         | <b>Without AES-NI:</b><br><b>CBC, ECB</b> ( e/d; 128 , 192 , 256 ); <b>CTR</b> ( enc; ext only; 128 , 192 , 256 )<br><b>KW<sup>1</sup></b> ( AE , AD , AES-128 , AES-192 , AES-256 , FWD , 128 , 256 , 192 , 320 , 1024 ) | <a href="#">C786</a> |
|                             | <b>With AES-NI:</b><br><b>CBC, ECB</b> ( e/d; 128 , 192 , 256 ); <b>CTR</b> ( ext only; 128 , 192 , 256 )<br><b>KW<sup>1</sup></b> ( AE , AD , AES-128 , AES-192 , AES-256 , FWD , 128 , 256 , 192 , 320 , 1024 )         | <a href="#">C794</a> |

<sup>1</sup> The AES key wrapping provides between 128 and 256 bits of encryption strength.

| Approved Security Functions       |  | Certificate          |
|-----------------------------------|--|----------------------|
| Triple-DES<br>(3Key) <sup>2</sup> | TCBC( KO 1 e/d, ) ; TECB ( KO 1 e/d, ) ; CTR ( enc; ext only )   | <a href="#">C786</a> |
| <b>Secure Hash Standard (SHS)</b> |  |                      |
| SHS                               | SHA-1 (BYTE-only)<br>SHA-224 (BYTE-only)<br>SHA-256 (BYTE-only)<br>SHA-384 (BYTE-only)<br>SHA-512 (BYTE-only)  | <a href="#">C786</a> |
| <b>Data Authentication Code</b>   |  |                      |
| HMAC                              | HMAC-SHA1 ( Key Size Ranges Tested: KS<BS KS=BS KS>BS )<br>HMAC-SHA224 ( Key Size Ranges Tested: KS<BS KS=BS KS>BS )<br>HMAC-SHA256 ( Key Size Ranges Tested: KS<BS KS=BS KS>BS )<br>HMAC-SHA384 ( Key Size Ranges Tested: KS<BS KS=BS KS>BS )<br>HMAC-SHA512 ( Key Size Ranges Tested: KS<BS KS=BS KS>BS )  | <a href="#">C786</a> |
| <b>Asymmetric Algorithms</b>      |  |                      |
| RSA                               | FIPS186-4:<br>186-4KEY(gen):<br>FIPS186-4_Fixed_e ( 10001 ); PGM(ProbRandom: ( 2048 , 3072 ) PPTT:( C.3 )<br>ALG[RSASSA-PKCS1_V1_5]:<br>SIG(gen) (2048 SHA(224, 256, 384, 512)) (3072 SHA(224, 256, 384, 512))<br>Sig(Ver) (1024 SHA( 1, 224, 256, 384, 512)) (2048 SHA(1, 224, 256, 384, 512)) (3072 SHA(1, 224, 256, 384, 512))  | <a href="#">C786</a> |
| DSA                               | FIPS186-4:<br>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 ) ]<br><br>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 ); ]<br>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 ) ] | <a href="#">C786</a> |
| ECDSA                             | FIPS186-4:<br>PKG: CURVES ( P-256 P-384 P-521 ExtraRandomBits )<br>PKV: CURVES ( P-256 P-384 P-521 )<br>SigGen: CURVES( P-256: (SHA-224, 256, 384, 512) P-384: (SHA-224, 256, 384, 512) P-521: (SHA-224, 256, 384, 512)<br>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) )  | <a href="#">C786</a> |
| <b>Random Number Generation</b>   |  |                      |
| DRBG                              | Hash_Based DRBG: [ Prediction Resistance Tested: Not Enabled ( SHA-256 )   | <a href="#">C786</a> |
| <b>Key Transport Scheme (KTS)</b> |  |                      |

<sup>2</sup> 3-Key Triple-DES key shall not be used to encrypt more than 2<sup>16</sup> 64-bit blocks of data.

| Approved Security Functions |  | Certificate          |
|-----------------------------|--|----------------------|
| AES-KW with AES-NI          | AES (128, 192, 256) key establishment methodology provides between 128 and 256 bits of encryption strength | <a href="#">C794</a> |
| AES-KW without AES-NI       | AES (128, 192, 256) key establishment methodology provides between 128 and 256 bits of encryption strength | <a href="#">C786</a> |

**Table 2: FIPS Approved Security Functions**

### 3.4 Non-Approved but Allowed Security Functions

The following are considered non-Approved but allowed security functions:

| Algorithm | Usage                                  |
|-----------|--|
| NDRNG     | Used for seeding NIST SP 800-90A DRBG. |

**Table 3: Non-Approved but Allowed Security Functions**

### 3.5 Non-Approved Security Functions

The following algorithms are considered non-Approved. Using any of these algorithms will put the module in the non-Approved mode implicitly. The services associated with these non-Approved algorithms are specified in Section 7.3.

| Algorithm                       | Usage   |
|---------------------------------|---|
| AES CTS mode                    | Encryption/Decryption   |
| AES GCM                         | Encryption/Decryption (non-compliant with IG A.5, CAVS tested with Cert. # C 786)   |
| Camellia                        | Encryption/Decryption   |
| DES                             | Encryption/Decryption   |
| RC2                             | Encryption/Decryption   |
| RC4                             | Encryption/Decryption   |
| RC5                             | Encryption/Decryption   |
| SEED                            | Encryption/Decryption   |
| Two-key Triple-DES              | Encryption/Decryption, key wrapping using two-key Triple-DES  |
| MD2                             | Hashing   |
| MD5                             | Hashing   |
| DSA with non-compliant key size | DSA key pair generation with key size less than 2048 bits. DSA signature generation with key size less than 2048 bits. DSA signature verification with key size less than 1024 bits.  |
| RSA with non-compliant key size | RSA key pair generation with key size less than 2048 bits. RSA signature generation with key size less than 2048 bits. RSA signature verification with key size less than 1024 bits.. |
| Diffie-Hellman                  | Key agreement.  |
| Elliptic Curve Diffie-Hellman   | Key agreement.  |
| J-PAKE                          | Key agreement   |

| Algorithm    | Usage  |
|--------------|--|
| TLS KDF      | Key derivation   |
| Key Wrapping | Non-SP 800-38F AES and Triple-DES Key wrapping, RSA key wrapping |

**Table 4: Non-Approved and Disallowed Functions**

## 4. Module Ports and Interfaces

The module FIPS 140 interfaces can be categorized as follows:

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

As a software-only module, the module does not have physical ports. For the purpose of FIPS 140-2 validation, the physical ports of the module are interpreted to be the physical ports of the hardware platform on which it runs. The logical interface is a C-language Application Program Interface (API) following the PKCS #11 specification, the database files in kernel file system, and environment variables.

The module uses different function arguments for input and output to distinguish between data input, control input, data output, and status output, to disconnect the logical paths followed by data/control entering the module and data/status exiting the module. The module doesn't use the same buffer for input and output. After the module is done with an input buffer that holds security related information, it always zeroizes the buffer so that if the memory is later reused as an output buffer, no sensitive information can be inadvertently leaked.

Table 5 below shows the mapping of interfaces as per FIPS 140-2 standard.

| FIPS 140-2 Interface | Module Interfaces  |
|----------------------|--|
| Data Input           | API input parameters and database files in kernel file system  |
| Data Output          | API output parameters and database files in kernel file system |
| Control Input        | API function calls and environment variables                   |
| Status Output        | API return codes and status parameters                         |

**Table 5: Mapping of FIPS 140-2 Logical Interfaces**

### 4.1 PKCS#11

The logical interfaces of the module consist of the PKCS #11 (Cryptoki) API. The API itself defines the module's logical boundary, i.e., all access to the module is through this API. The functions in the PKCS #11 API are listed in Table 8.

### 4.2 Inhibition of Data Output

All data output via the data output interface is inhibited when the NSS cryptographic module is performing self-tests or in the Error state.

- During self-tests: All data output via the data output interface is inhibited while self-tests are executed.
- In Error state: The Boolean state variable `sftk_fatalError` tracks whether the NSS cryptographic module is in the Error state. Most PKCS #11 functions, including all the functions that output data via the data output interface, check the `sftk_fatalError` state variable and, if it is true, return the `CKR_DEVICE_ERROR` error code immediately. Only the functions that shut down and restart the module, reinitialize the module, or output status information can be invoked in the Error state. These functions are `FC_GetFunctionList`, `FC_Initialize`, `FC_Finalize`, `FC_GetInfo`, `FC_GetSlotList`, `FC_GetSlotInfo`, `FC_GetTokenInfo`, `FC_InitToken`, `FC_CloseSession`, `FC_CloseAllSessions`, and `FC_WaitForSlotEvent`.



### **4.3 Disconnecting the Output Data Path from the Key Processes**

During key generation and key zeroization, the module may perform audit logging, but the audit records do not contain sensitive information. The module does not return the function output arguments until the key generation or key zeroization is finished. Therefore, the logical paths used by output data exiting the module are logically disconnected from the processes/threads performing key generation and key zeroization.



## 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 module operates in a modifiable operational environment under the FIPS 140-2 definition. The module was tested on the following environments:

| Operating Environment   | Processor                | Hardware           | With or without PAA |
|-------------------------|--------------------------|--------------------|---------------------|
| Oracle Linux 7.6 64-bit | Intel® Xeon® Silver 4114 | Oracle Server X7-2 | With PAA (AES-NI)   |
| Oracle Linux 7.6 64-bit | Intel® Xeon® Silver 4114 | Oracle Server X7-2 | Without PAA         |
| Oracle Linux 7.6 64-bit | AMD® EPYC® 7551          | Oracle Server X7-2 | With PAA (AES-NI)   |
| Oracle Linux 7.6 64-bit | AMD® EPYC® 7551          | Oracle Server X7-2 | Without PAA         |

**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 |



| Operating Environment   | Processor                          | Hardware                       |
|-------------------------|------------------------------------|--------------------------------|
| 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      |
| 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 PRIMEQUEST 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 PRIMERGY 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     |

| Operating Environment   | Processor   | Hardware                                      |
|-------------------------|---|---|
| Oracle Linux 7.6 64-bit | Intel® Xeon® Scalable Processors  | Fujitsu PRIMERGY RX4770 M4                    |
| 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 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 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 | Intel® Xeon® E5-4600 v3   | HPE ProLiant BL660c 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® 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® E5-4600/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® 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® E5-2600 v3 & v4  | HPE ProLiant ML350 Gen9                       |
| 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® E7-4800 v4/E7-8800 v4  | HPE Synergy 620 Gen9 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                  |

| Operating Environment   | Processor   | Hardware                       |
|-------------------------|---|--------------------------------|
| 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® E5-2600 v3 & v4                              | Inspur Yingxin NF5240M4        |
| 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® 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® 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® 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             |

| Operating Environment   | Processor                                       | Hardware                               |
|-------------------------|---|--|
| 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 Sun 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 A840-G10                         |
| Oracle Linux 7.6 64-bit | Intel® Xeon® E5-2600 v3 & v4                    | Sugon CB50-G20                         |
| Oracle Linux 7.6 64-bit | Intel® Xeon® E7-4800 v2                         | Sugon CB80-G20                         |
| Oracle Linux 7.6 64-bit | Intel® Xeon® E7-4800 v4                         | 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® 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 Operational Environments**

*Note:* 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 (concurrent operators are explicitly excluded).

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



In operational mode, the ptrace system call, the debugger gdb, and strace shall be not used. In addition, other tracing mechanisms offered by the Linux environment, such as ftrace or systemtap, shall not be used.

## 7. Roles, Services and Authentication

This section defines the roles, services, and authentication mechanisms and methods with respect to the applicable FIPS 140-2 requirements.

### 7.1 Roles

The module implements a Crypto Officer (CO) role and a User role:

- The CO role is supported for the installation and initialization of the module. Also, the CO role can access other general-purpose services (such as message digest and random number generation services) and status services of the module. The CO does not have access to any service that utilizes the secret or private keys of the module. The CO must control the access to the module both before and after installation, including management of physical access to the computer, executing the module code as well as management of the security facilities provided by the operating system.
- The User role has access to all cryptographically secure services which use the secret or private keys of the module. It is also responsible for the retrieval, updating and deletion of keys from the private key database.

### 7.2 FIPS Approved Operator Services and Descriptions

The module has a set of API functions denoted by FC\_xxx as listed in Table 8. Among the module's API functions, only FC\_GetFunctionList is exported and therefore callable by its name. All the other API functions must be called via the function pointers returned by FC\_GetFunctionList. It returns a CK\_FUNCTION\_LIST structure containing function pointers named C\_xxx such as C\_Initialize and C\_Finalize. The C\_xxx function pointers in the CK\_FUNCTION\_LIST structure returned by FC\_GetFunctionList point to the FC\_xxx functions.

The following convention is used to describe API function calls. Here FC\_Initialize is used as an example:

- When “call FC\_Initialize” is mentioned, the technical equivalent of “call the FC\_Initialize function via the C\_Initialize function pointer in the CK\_FUNCTION\_LIST structure returned by FC\_GetFunctionList” is implied.

The module supports Crypto-Officer services which require no operator authentication, and User services which require operator authentication. Crypto-Officer services do not require access to the secret and private keys and other CSPs associated with the user. The message digesting services are available to Crypto-Officer only when CSPs are not accessed. User services which access CSPs (e.g., FC\_GenerateKey, FC\_GenerateKeyPair) require operator authentication.

Table 8 lists all the services available in FIPS Approved mode. Please refer to Table 2 and Table 3 for the Approved or allowed cryptographic algorithms supported by the module.

| U | CO | Service Name              | Function             | Service Description  | Keys and CSP(s)            | Access |
|---|----|---------------------------|----------------------|--|----------------------------|--------|
|   | X  | Get Function List         | FC_GetFunctionList   | Return a pointer to the list of function pointers for the operational mode   | None                       | -      |
|   | X  | Module Initialization     | FC_InitToken         | Initialize or re-initialize a token  | User password and all keys | Z      |
|   | X  |                           | FC_InitPIN           | Initialize the user's password, i.e., set the user's initial password  | User password              | W      |
|   | X  | General Purpose           | FC_Initialize        | Initialize the module library  | None                       | -      |
|   | X  |                           | FC_Finalize          | Finalize (shut down) the module library  | All keys                   | Z      |
|   | X  |                           | FC_GetInfo           | Obtain general information about the module library  | None                       | -      |
|   | X  | Slot and Token Management | FC_GetSlotList       | Obtain a list of slots in the system   | None                       | -      |
|   | X  |                           | FC_GetSlotInfo       | Obtain information about a particular slot   | None                       | -      |
|   | X  |                           | FC_GetTokenInfo      | Obtain information about the token (This function provides the Show Status service)  | None                       | -      |
|   | X  |                           | FC_GetMechanismList  | Obtain a list of mechanisms (cryptographic algorithms) supported by a token  | None                       | -      |
|   | X  |                           | FC_GetMechanismInfo  | Obtain information about a particular mechanism  | None                       | -      |
| X |    |                           | FC_SetPIN            | Change the user's password   | User password              | R, W   |
|   | X  | Session Management        | FC_OpenSession       | Open a connection (session) between an application and a particular token  | None                       | -      |
|   | X  |                           | FC_CloseSession      | Close a session  | All keys for the session   | Z      |
|   | X  |                           | FC_CloseAllSessions  | Close all sessions with a token  | All keys                   | Z      |
|   | X  |                           | FC_GetSessionInfo    | Obtain information about the session (This function provides the Show Status service)  | None                       | -      |
|   | X  |                           | FC_GetOperationState | Save the state of the cryptographic operations in a session (This function is only implemented for message digest operations)    | None                       | -      |
|   | X  |                           | FC_SetOperationState | Restore the state of the cryptographic operations in a session (This function is only implemented for message digest operations) | None                       | -      |

| U | CO | Service Name                          | Function                  | Service Description  | Keys and CSP(s)                     | Access             |
|---|----|---------------------------------------|---------------------------|--|-------------------------------------|--------------------|
| X |    |                                       | FC_Login                  | Log into a token   | User Password                       | R, W, X            |
| X |    |                                       | FC_Logout                 | Log out from a token   | None                                | -                  |
| X |    | Object Management                     | FC_CreateObject           | Create a new object  | Key                                 | W                  |
| X |    |                                       | FC_CopyObject             | Create a copy of an object   | Original Key, new key <sup>3</sup>  | R, W               |
| X |    |                                       | FC_DestroyObject          | Destroy an object  | Key                                 | Z                  |
| X |    |                                       | FC_GetObjectSize          | Obtain the size of an object in bytes  | Key                                 | R                  |
| X |    |                                       | FC_GetAttributeValue      | Obtain an attribute value of an object   | Key                                 | R                  |
| X |    |                                       | FC_SetAttributeValue      | Modify an attribute value of an object   | Key                                 | W                  |
| X |    |                                       | FC_FindObjectsInit        | Initialize an object search operation  | None                                | -                  |
| X |    |                                       | FC_FindObjects            | Continue an object search operation  | Keys matching the search criteria   | R                  |
| X |    |                                       | FC_FindObjectsFinal       | Finish an object search operation  | None                                | -                  |
| X |    |                                       | Encryption and Decryption | FC_EncryptInit   | Initialize an encryption operation  | AES/Triple-DES key |
| X |    | FC_Encrypt                            |                           | Encrypt single-part data   | AES/Triple-DES key                  | R                  |
| X |    | FC_EncryptUpdate                      |                           | Continue a multiple-part encryption operation  | AES/Triple-DES key                  | R                  |
| X |    | FC_EncryptFinal                       |                           | Finish a multiple-part encryption operation  | AES/Triple-DES key                  | R                  |
| X |    | FC_DecryptInit                        |                           | Initialize a decryption operation  | AES/Triple-DES key                  | R                  |
| X |    | FC_Decrypt                            |                           | Decrypt single-part encrypted data   | AES/Triple-DES key                  | R                  |
| X |    | FC_DecryptUpdate                      |                           | Continue a multiple-part decryption operation  | AES/Triple-DES key                  | R                  |
| X |    | FC_DecryptFinal                       |                           | Finish a multiple-part decryption operation  | AES/Triple-DES key                  | R                  |
|   | X  | Message Digest                        | FC_DigestInit             | Initialize a message digesting operation   | None                                | -                  |
|   | X  |                                       | FC_Digest                 | Digest single-part data  | None                                | -                  |
|   | X  |                                       | FC_DigestUpdate           | Continue a multiple-part digesting operation   | None                                | -                  |
| X |    |                                       | FC_DigestKey              | Continue a multiple-part message-digesting operation by digesting the value of a secret key as part of the data already digested | HMAC Key                            | R                  |
|   | X  |                                       | FC_DigestFinal            | Finish a multiple-part digesting operation   | None                                | -                  |
| X |    | Signature Generation and Verification | FC_SignInit               | Initialize a signature operation   | DSA/ECDSA/RSA private key, HMAC key | R                  |

<sup>3</sup> 'Original key' and 'New key' are the secret keys or public/private key pairs.



| U | CO | Service Name           | Function                               | Service Description   | Keys and CSP(s)   | Access             |
|---|----|------------------------|--|---|---|--------------------|
| X |    |                        | FC_Sign                                | Sign single-part data   | DSA/ECDSA/RSA private key, HMAC key                         | R                  |
| X |    |                        | FC_SignUpdate                          | Continue a multiple-part signature operation  | DSA/ECDSA/RSA private key, HMAC key                         | R                  |
| X |    |                        | FC_SignFinal                           | Finish a multiple-part signature operation  | DSA/ECDSA/RSA private key, HMAC key                         | R                  |
| X |    |                        | FC_SignRecoverInit                     | Initialize a signature operation, where the data can be recovered from the signature        | DSA/ECDSA/RSA private key                                   | R                  |
| X |    |                        | FC_SignRecover                         | Sign single-part data, where the data can be recovered from the signature                   | DSA/ECDSA/RSA private key                                   | R                  |
| X |    |                        | FC_VerifyInit                          | Initialize a verification operation   | DSA/ECDSA/RSA public key, HMAC key                          | R                  |
| X |    |                        | FC_Verify                              | Verify a signature on single-part data  | DSA/ECDSA/RSA public key, HMAC key                          | R                  |
| X |    |                        | FC_VerifyUpdate                        | Continue a multiple-part verification operation   | DSA/ECDSA/RSA public key, HMAC key                          | R                  |
| X |    |                        | FC_VerifyFinal                         | Finish a multiple-part verification operation   | DSA/ECDSA/RSA public key, HMAC key                          | R                  |
| X |    |                        | FC_VerifyRecoverInit                   | Initialize a verification operation, where the data is recovered from the signature         | DSA/ECDSA/RSA public key                                    | R                  |
| X |    |                        | FC_VerifyRecover                       | Verify a signature on single-part data, where the data is recovered from the signature      | DSA/ECDSA/RSA public key                                    | R                  |
| X |    |                        | Dual Function Cryptographic Operations | FC_DigestEncryptUpdate  | Continue a multiple-part digesting and encryption operation | AES/Triple-DES key |
| X |    | FC_DecryptDigestUpdate |  | Continue a multiple-part decryption and digesting operation                                 | AES/Triple-DES key  | R                  |
| X |    | FC_SignEncryptUpdate   |  | Continue a multiple-part signing and encryption operation                                   | DSA/ECDSA/RSA private key, HMAC key, AES/Triple-DES key     | R                  |
| X |    | FC_DecryptVerifyUpdate |  | Continue a multiple-part decryption and verify operation                                    | DSA/ECDSA/RSA public key, HMAC key, AES/Triple-DES key      | R                  |
| X |    | Key Management         | FC_GenerateKey                         | Generate a secret key   | AES/Triple-DES/HMAC key                                     | W                  |
|   |    |                        | FC_GenerateKeyPair                     | Generate a public/private key pair (This function performs the pair-wise consistency tests) | DSA/ECDSA/RSA key pair                                      | W                  |

| U | CO | Service Name                 | Function                               | Service Description  | Keys and CSP(s)  | Access |
|---|----|------------------------------|--|--|--|--------|
|   |    |                              | FC_WrapKey                             | Wrap (encrypt) a key using one of the following mechanisms:<br>(1) SP 800-38F AES Key wrapping     | Wrapping Key <sup>4</sup> , Key to be wrapped <sup>5</sup> | R      |
|   |    |                              | FC_UnwrapKey                           | Unwrap (decrypt) a key using one of the following mechanisms:<br>(1) SP 800-38F AES Key unwrapping | Unwrapping key <sup>6</sup>                                | R      |
|   |    |                              |  |  | Unwrapped key <sup>7</sup>                                 | W      |
|   | X  | Random Number Generation     | FC_SeedRandom                          | Mix in additional seed material to the random number generator                                     | Entropy input string, seed, DRBG V and C values            | R, W   |
|   |    |                              | FC_GenerateRandom                      | Generate random data (This function performs the continuous random number generator test)          | Random data, DRBG V and C values                           | R, W   |
|   | X  | Parallel Function Management | FC_GetFunctionStatus                   | A legacy function, which simply returns the value 0x00000051 (function not parallel)               | None   | -      |
|   |    |                              | FC_CancelFunction                      | A legacy function, which simply returns the value 0x00000051 (function not parallel)               | None   | -      |
|   | X  | Self-Tests                   | N/A                                    | The self-tests are performed automatically when loading the module                                 | DSA 2048-bit public key for module integrity test          | R      |
| X |    | Zeroization                  | FC_DestroyObject                       | All CSPs are automatically zeroized when freeing the cipher handle                                 | All secret or private keys and password                    | Z      |
|   | X  |                              | FC_InitToken                           |  |  |        |
|   |    |                              | FC_Finalize                            |  |  |        |
|   |    |                              | FC_CloseSession<br>FC_CloseAllSessions |  |  |        |

**Table 8: FIPS Approved Operator Services and Descriptions**

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

<sup>4</sup> 'Wrapping key' corresponds to the secret key or public key used to wrap another key.

<sup>5</sup> 'Key to be wrapped' is the key that is wrapped by the 'wrapping key'.

<sup>6</sup> 'Unwrapping key' corresponds to the secret key or private key used to unwrap another key.

<sup>7</sup> 'Unwrapped key' is the plaintext key that has not been wrapped by a 'wrapping key'.

*Note:* The message digesting functions (except FC\_DigestKey) that do not use any keys of the module can be accessed by the Crypto-Officer role and do not require authentication to the module. The FC\_DigestKey API function computes the message digest (hash) of the value of a secret key, so it is available only to the User role.

### 7.3 Non-FIPS Approved Services and Descriptions

Table 9 lists all the services available in non-Approved mode with API function and the non- Approved algorithm that the function may invoke. Please note that the functions are the same as the ones listed in

**Table 8**, but the underneath non-Approved algorithms are invoked. Please also refer to Table 4 for the non-Approved algorithms. If any service invokes the non-Approved algorithms, then the module will enter non-Approved mode implicitly.

| Service Name                          | Function           | Non-Approved Algorithm Invoked   |
|---------------------------------------|--------------------|--|
| Encryption and Decryption             | FC_EncryptInit     | AES-GCM mode, AES-CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, Two-Key Triple-DES   |
|                                       | FC_Encrypt         |  |
|                                       | FC_EncryptUpdate   |  |
|                                       | FC_EncryptFinal    |  |
|                                       | FC_DecryptInit     | AES-GCM mode, AES-CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, Two-key Triple-DES   |
|                                       | FC_Decrypt         |  |
|                                       | FC_DecryptUpdate   |  |
|                                       | FC_DecryptFinal    |  |
| Message Digest                        | FC_DigestInit      | MD2, MD5   |
|                                       | FC_Digest          |  |
|                                       | FC_DigestUpdate    |  |
|                                       | FC_DigestKey       |  |
|                                       | FC_DigestFinal     |  |
| Signature Generation and Verification | FC_SignInit        | DSA signature generation with non-compliant key size < 2048, RSA signature generation with non-compliant key sizes < 2048      |
|                                       | FC_Sign            |  |
|                                       | FC_SignUpdate      |  |
|                                       | FC_SignFinal       |  |
|                                       | FC_SignRecoverInit |  |
|                                       | FC_SignRecover     |  |
|                                       | FC_VerifyInit      | DSA signature verification with non-compliant key sizes < 1024, RSA signature verification with non-compliant key sizes < 1024 |
|                                       | FC_Verify          |  |

| Service Name                           | Function               | Non-Approved Algorithm Invoked   |
|--|------------------------|--|
|  | FC_VerifyUpdate        |  |
|  | FC_VerifyFinal         |  |
|  | FC_VerifyRecoverInit   |  |
|  | FC_VerifyRecover       |  |
| Dual Function Cryptographic Operations | FC_DigestEncryptUpdate | MD2, MD5, AES-GCM mode, AES-CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, Two-Key Triple-DES   |
|  | FC_DecryptDigestUpdate | AES-GCM mode, AES-CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, MD2, MD5   |
|  | FC_SignEncryptUpdate   | DSA signature generation with non-compliant key sizes < 2048, RSA signature generation with non-compliant key sizes < 2048, AES-GCM mode, AES CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, Two-key Triple-DES     |
|  | FC_DecryptVerifyUpdate | DSA signature verification with non-compliant key sizes < 1024, RSA signature verification with non-compliant key sizes < 1024, AES-GCM mode, AES-CTS mode, Camellia, DES, RC2, RC4, RC5, SEED, Two-key Triple-DES |
| Key Management                         | FC_GenerateKey         | TLS Pre-Master Secret generation.  |
|  | FC_GenerateKeyPair     | DSA key pair generation with non-compliant key sizes < 2048, RSA key pair generation with non-compliant key sizes < 2048, Diffie-Hellman key pair generation, EC Diffie-Hellman key pair generation.               |
|  | FC_WrapKey             | Triple-DES key wrapping (encrypt) using Two-key or Three-key Triple-DES, RSA key wrapping (encrypt), non-SP 800-38F AES key wrapping (encrypt)   |
|  | FC_UnwrapKey           | Triple-DES key wrapping (decrypt) using Two-key or Three-key Triple-DES, RSA key wrapping (decrypt) , non-SP 800-38F AES key unwrapping (decrypt)  |
|  | FC_DeriveKey           | Diffie-Hellman key agreement, Elliptic Curve Diffie-Hellman key agreement, J-PAKE key agreement  |

**Table 9: Non-FIPS Approved Operator Services and Descriptions**

## 7.4 Operator Authentication

### 7.4.1 Role Assumption

The CO role is implicitly assumed by an operator while installing the module by following the instructions in Section 10.1 and while performing other CO services on the module.

The module implements a password-based Role based authentication for the User role as defined by FIPS 140-2. To perform any security services under the User role, an operator must log into the module and complete an authentication procedure using the password information unique to the User role operator. The password is passed to the module via the API function as an input argument and won't be displayed. The return value of the function is the only feedback mechanism, which does not provide any information that could be used to guess or determine the User's password. The password is initialized by the CO role as part of module initialization and can be changed by the User role operator.

If a User-role service is called before the operator is authenticated, it returns the CKR\_USER\_NOT\_LOGGED\_IN error code. The operator must call the FC\_Login function to provide the required authentication.

Once a password has been established for the module, the user is allowed to use the security services if and only if the user is successfully authenticated to the module. Password establishment and authentication are required for the operation of the module. When the module is powered off, the result of previous authentication will be cleared and the user needs to be re-authenticated.

## 7.4.2 Strength of Authentication Mechanism

The module imposes the following requirements on the password. These requirements are enforced by the module on password initialization or change.

- The password must be at least seven characters long.
- The password must consist of characters from three or more character classes. We define five character classes: digits (0-9), ASCII lowercase letters (a-z), ASCII uppercase letters (AZ), ASCII non-alphanumeric characters (space and other ASCII special characters such as '\$', '!'), and non-ASCII characters (Latin characters such as 'e', 's'; Greek characters such as 'Ω', 'θ'; other non-ASCII special characters such as '.'). If an ASCII uppercase letter is the first character of the password, the uppercase letter is not counted toward its character class. Similarly, if a digit is the last character of the password, the digit is not counted toward its character class.

To estimate the maximum probability that a random guess of the password will succeed, we assume that:

- The characters of the password are independent with each other.
- The password contains the smallest combination of the character classes, which are five digits, one ASCII lowercase letter and one ASCII uppercase letter. The probability to guess every character successfully is  $(1/10)^5 * (1/26) * (1/26) = 1/67,600,000$ .

Since the password can contain seven characters from any three or more of the aforementioned five character classes, the probability that a random guess of the password will succeed is less than or equals to  $1/67,600,000$ , which is smaller than the required threshold  $1/1,000,000$ .

After each failed authentication attempt, the NSS cryptographic module inserts a one-second delay before returning to the caller, allowing at most 60 authentication attempts during a one-minute period. Therefore, the probability of a successful random guess of the password during a one-minute period is less than or equals to  $60 * 1/67,600,000 = 0.089 * (1/100,000)$ , which is smaller than the required threshold  $1/100,000$ .

## 8. Key and CSP Management

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

| CSP Name                                | Generation                                | Entry/Output   | Storage   | Zeroization   |
|---|---|--|---|---|
| AES Key (128, 192, 256 bits)            | NIST SP 800-90A DRBG                      | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| Triple-DES Key (192 bits)               | NIST SP 800-90A DRBG                      | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| DSA Private Key (2048 and 3072 bits)    | FIPS 186-4                                | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| ECDSA Private Key (P-256, P-384, P-521) | FIPS 186-4                                | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| RSA Private Key (2048, 3072, 4096 bits) | FIPS 186-4                                | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| HMAC Key (≥ 112 bits)                   | NIST SP 800-90A DRBG                      | Encrypted through key wrapping using FC_WrapKey <sup>8</sup> | Application memory or key database                | Automatically zeroized when freeing the cipher handle   |
| DRBG Entropy Input String and Seed      | Obtained from NDRNG                       | N/A  | Application memory                                | Automatically zeroized when freeing DRBG handle   |
| DRBG V and C values                     | Derived from entropy input                | N/A  | Application memory                                | Automatically zeroized when freeing DRBG handle   |
| User Passwords                          | N/A (supplied by the calling application) | N/A (input through API parameter)                            | Application memory or key database in salted form | Automatically zeroized when the module is reinitialized or overwritten when the user changes its password |

**Table 10: CSP Table**

### 8.1 Random Number Generation

The module employs a NIST SP800-90A Hash\_DRBG with SHA-256 to generate symmetric keys, HMAC keys. For symmetric keys, the generated key is an unmodified output from a DRBG. For generating asymmetric keys, the module implements key generation services compliant with FIPS186-4, and the seed (i.e., the random value) used in asymmetric key generation is obtained from SP800-90A DRBG unmodified. In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133 (vendor affirmed).

<sup>8</sup> Key wrapping using a FIPS Approved symmetric key with equal or greater bits of security strength



The module uses NDRNG from `/dev/urandom` as a source of entropy for seeding the DRBG. The NDRNG is provided by the operating environment (i.e., Linux RNG), which is within the module's physical boundary but outside of its logical boundary. The NDRNG provides at least 130 bits of entropy to the DRBG.

*CAVEAT:* The module generates cryptographic keys whose strengths are modified by available entropy.

Reseeding is performed by pulling more data from `/dev/urandom`. A product using the module should periodically reseed the module's DRBG with unpredictable noise by calling `FC_SeedRandom`. After  $2^{48}$  calls to the random number generator the module reseeds automatically. The module performs DRBG health testing as specified in Section 11.3 of NIST SP800-90A and the continuous random number generator test (CRNGT) on the output of DRBG to ensure that consecutive random numbers do not repeat. The underlying operating system performs the continuous test on the NDRNG. If CRNGT fails, the operating system kernel panics, and the module is not available for use.

## 8.2 Key/CSP Storage

The module employs the cryptographic keys and CSPs in the FIPS Approved mode of operation as listed in Table 10. The module does not perform persistent storage for any keys or CSPs. Note that the private key database (provided with the files `key3.db/key4.db`) mentioned in Table 10 is within the module's physical boundary but outside its logical boundary.

## 8.3 Key/CSP Zeroization

The application that uses the module is responsible for appropriate zeroization of the key material. The module provides zeroization methods to clear the memory region previously occupied by a plaintext secret key, private key or password. A plaintext secret or private key gets zeroized when it is passed to an `FC_DestroyObject` call. All plaintext secret and private keys must be zeroized when the module is shut down (with an `FC_Finalize` call), reinitialized (with an `FC_InitToken` call), or when the session is closed (with an `FC_CloseSession` or `FC_CloseAllSessions` call). All zeroization is to be performed by storing the value 0 into every byte of the memory region that is previously occupied by a plaintext secret key, private key or password. Zeroization is performed in a time that is not sufficient to compromise plaintext secret or private keys and password.

## 9. Self-Tests

FIPS 140-2 requires that the module perform 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.

### 9.1 Power-Up Self-Tests

All the power-up self-tests are performed automatically by initializing or re-initializing the module without requiring any operator intervention. During the power-up self-tests, no cryptographic operation is available and all input or output is inhibited. Once the power-up self-tests are completed successfully, the module enters operational mode and cryptographic operations are available. If any of the power-up self-tests fail, the module enters the Error state. In Error state, all output is inhibited and no cryptographic operation is allowed. The module returns the error code CKR\_DEVICE\_ERROR to the calling application to indicate the Error state. The module needs to be reinitialized in order to recover from the Error state. The following table provides the lists of Known-Answer Test (KAT) and Integrity Test as the power up self-tests:

| Algorithm        | Test   |
|------------------|--|
| AES              | KATs for ECB and CBC modes: encryption and decryption are tested separately  |
| Triple-DES       | KATs for ECB and CBC modes: encryption (for legacy purposes) and decryption are tested separately  |
| DSA              | KAT: signature generation and verification are tested separately   |
| ECDSA            | KAT: signature generation and verification are tested separately   |
| RSA              | KAT: encryption and decryption are tested separately (both are used for legacy purposes)<br>KAT: signature generation and verification are tested separately |
| SHA              | KAT: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512   |
| HMAC             | KAT: HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512  |
| DRBG             | KAT of Hash_DRBG with SHA-256  |
| Module Integrity | DSA signature verification with 2048 bits key and SHA-256  |

**Table 11: Power-On Self-Tests**

The power-up self-tests can be performed on demand by reinitializing the module.

### 9.2 Conditional Self-Tests

The following table provides the lists of Pairwise Consistency Test (PCT) and Continuous Random Number Generation Test (CRNGT) as the conditional self-tests. If any of the conditional test fails, the module enters the Error state. It returns the error code CKR\_DEVICE\_ERROR to the calling application to indicate the Error state. The module needs to be reinitialized in order to recover from the Error state.

| Algorithm | Test   |
|-----------|--|
| DSA       | PCT for DSA key generation                                     |
| ECDSA     | PCT for ECDSA key generation                                   |
| RSA       | PCT for RSA key generation                                     |
| DRBG      | CRNGT  |
| DRBG      | Health testing as specified in Section 11.3 of NIST SP 800-90A |

**Table 12: Conditional Self-Tests**



## 10. Crypto-Officer and User Guidance

### 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 shall 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 shall also verify the hash of the RPM package to confirm a proper download.

In addition, to support the Module, the NSPR library shall be installed that is offered by the underlying operating system.

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

1. Ensure 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:
  - 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> shall be supplied. The partition can be identified with the command:

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

| <u>Filesystem</u> | <u>1K-blocks</u> | <u>Used</u> | <u>Available</u> | <u>Use%</u> | <u>Mounted on</u> |
|-------------------|------------------|-------------|------------------|-------------|-------------------|
| /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 operating system kernel validation checks, where the kernel will be verified 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 shall 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 in the Operating System 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.

If an application that uses the Module for its cryptography is put into a chroot environment, the Crypto Officer shall ensure one of the above methods is available to the Module from within the chroot environment to ensure entry into FIPS Approved mode. Failure to do so will not allow the application to properly enter FIPS Approved mode.

### 10.1.1 Access to Audit Data

The module may use the Unix syslog function and the audit mechanism provided by the operating system to audit events. Auditing is turned off by default. Auditing capability shall be turned on as part of the initialization procedures by setting the environment variable `NSS_ENABLE_AUDIT` to 1. The Crypto-Officer shall also configure the operating system's audit mechanism.

The module uses the syslog function to audit events, so the audit data are stored in the system log. Only the root user can modify the system log. On some platforms, only the root user can read the system log; on other platforms, all users can read the system log. The system log is usually under the `/var/log` directory. The exact location of the system log is specified in the `/etc/syslog.conf` file. The module uses the default user facility and the info, warning, and err severity levels for its log messages.

The module can also be configured to use the audit mechanism provided by the operating system to audit events. The audit data would then be stored in the system audit log. Only the root user can read or modify the system audit log. To turn on this capability it is necessary to create a symbolic link from the library file `/usr/lib64/libaudit.so.0` to `/usr/lib64/libaudit.so.1.0.0` (on 64-bit platforms).

## 10.2 User Guidance

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

The following module initialization steps shall be followed before starting to use the NSS module:

- Set the environment variable `NSS_ENABLE_AUDIT` to 1 before using the module with an application.
- Use the application to get the function pointer list using the API “`FC_GetFunctionList`”.



- Use the API FC\_Initialize to initialize the module and ensure that it returns CKR\_OK. A return code other than CKR\_OK means the module is not initialized correctly, and in that case, the module shall be reset and initialized again.
- For the first login, provide a NULL password and login using the function pointer C\_Login, which will in-turn call FC\_Login API of the module. This is required to set the initial NSS User password.
- Now, set the initial NSS User role password using the function pointer C\_InitPIN. This will call the module's API FC\_InitPIN API. Then, logout using the function pointer C\_Logout, which will call the module's API FC\_Logout.
- The NSS User role can now be assumed on the module by logging in using the User password. And the Crypto-Officer role can be implicitly assumed by performing the Crypto-Officer services as listed in Section 7.2.

The module can be configured to use different private key database formats: key3.db or key4.db. “key3.db” format is based on the Berkeley DataBase engine and should not be used by more than one process concurrently. “key4.db” format is based on SQL DataBase engine and can be used concurrently by multiple processes. Both databases are considered outside the module's logical boundary and all data stored in these databases is considered stored in plaintext. The interface code of the module that accesses data stored in the database is considered part of the cryptographic boundary.

Secret and private keys, plaintext passwords and other security-relevant data items are maintained under the control of the cryptographic module. Secret and private keys shall be passed to the calling application in encrypted (wrapped) form with FC\_WrapKey and entered from calling application in encrypted form with FC\_UnwrapKey. The key transport methods allowed for this purpose in FIPS Approved mode are SP 800-38F based AES key wrapping and RSA key wrapping using the corresponding Approved modes and key sizes. *Note:* If the secret and private keys passed to the calling application are encrypted using a symmetric key algorithm, the encryption key may be derived from a password. In such a case, they should be considered to be in plaintext form.

Automated key transport methods shall use FC\_WrapKey and FC\_UnwrapKey to output or input secret and private keys from or to the module.

All cryptographic keys used in the FIPS Approved mode of operation shall be generated in the FIPS Approved mode or imported while running in the FIPS Approved mode.

### 10.2.1 TLS Operations

The module does not implement the TLS protocol. In the non-approved mode, the module implements the cryptographic operations, including TLS-specific key generation and derivation operations, which can be used to implement the TLS protocol.

### 10.2.2 RSA and DSA Keys

The module allows the use of 1024 bits RSA and DSA keys for legacy purposes including signature generation, which is disallowed to be used in FIPS Approved mode as per NIST SP800-131A.

Therefore, the cryptographic operations with the non-approved key sizes will result the module operating in non-Approved mode implicitly.



### **10.2.3 Triple-DES keys**

According to IG A.13, the same Triple-DES key shall not be used to encrypt more than  $2^{16}$  64-bit blocks of data. Encrypting greater than  $2^{16}$  blocks will result in the module operating in non-Approved mode implicitly. It is the the User's responsible for ensuring the module's compliance with this requirement

### **10.3 Handling Self-Test Errors**

When the module enters the Error state, it needs to be reinitialized to resume normal operation. Reinitialization is accomplished by calling FC\_Finalize followed by FC\_Initialize.

## 11. Mitigation of Other Attacks

The module is designed to mitigate the following attacks.

| Attack   | Mitigation Mechanism  | Specific Limit  |
|--|---|---|
| Timing attacks on RSA  | <p><b>RSA blinding</b></p> <p>Timing attack on RSA was first demonstrated by Paul Kocher in 1996 [15], who contributed the mitigation code to our module.</p> <p>Most recently Boneh and Brumley [16] showed that RSA blinding is an effective defense against timing attacks on RSA.</p> | None.   |
| Cache-timing attacks on the modular exponentiation operation used in RSA and DSA | <p><b>Cache invariant modular exponentiation</b></p> <p>This is a variant of a modular exponentiation implementation that Colin Percival [17] showed to defend against cache-timing attacks</p>   | This mechanism requires intimate knowledge of the cache line sizes of the processor. The mechanism may be ineffective when the module is running on a processor whose cache line sizes are unknown. |
| Arithmetic errors in RSA signatures  | <p><b>Double-checking RSA signatures</b></p> <p>Arithmetic errors in RSA signatures might leak the private key. Ferguson and Schneier [18] recommend that every RSA signature generation should verify the signature just generated.</p>  | None.   |

**Table 13: Mitigation of Other Attacks**

## Acronyms, Terms and Abbreviations

| Term   | Definition  |
|--------|---|
| AES    | Advanced Encryption Standard                        |
| AES-NI | Intel Advanced Encryption Standard New Instructions |
| CBC    | Cipher Block Chaining                               |
| CMVP   | Cryptographic Module Validation Program             |
| CCCS   | Canadian Centre for Cyber Security                  |
| CSP    | Critical Security Parameter                         |
| CTR    | Counter Block Chaining                              |
| CVL    | Component Validation List                           |
| DES    | Data Encryption Standard                            |
| DSA    | Digital Signature Algorithm                         |
| ECB    | Electronic Code Book                                |
| DRBG   | Deterministic Random Bit Generator                  |
| ECDSA  | Elliptic Curve Digital Signature Algorithm          |
| GCM    | Galois/Counter Mode                                 |
| HMAC   | (Keyed) Hash Message Authentication Code            |
| MAC    | Message Authentication Code                         |
| KAT    | Known Answer Test                                   |
| KDF    | Key Derivation Function                             |
| NIST   | National Institute of Standards and Technology      |
| NSS    | Network Security Services                           |
| PKCS   | Public-Key Cryptographic Standard                   |
| PUB    | Publication   |
| RSA    | Rivest, Shamir, Addleman                            |
| SHA    | Secure Hash Algorithm                               |
| TLS    | Transport Layer Security                            |

**Table 14: 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.

- [1] FIPS 140-2 Standard, <http://csrc.nist.gov/groups/STM/cmvp/standards.html>
- [2] FIPS 140-2 Implementation Guidance, <http://csrc.nist.gov/groups/STM/cmvp/standards.html>
- [3] FIPS 140-2 Derived Test Requirements, <http://csrc.nist.gov/groups/STM/cmvp/standards.html>
- [4] FIPS 197 Advanced Encryption Standard, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [5] FIPS 180-4 Secure Hash Standard, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), <http://csrc.nist.gov/publications/PubsFIPS.html>
- [7] FIPS 186-4 Digital Signature Standard (DSS), <http://csrc.nist.gov/publications/PubsFIPS.html>
- [8] NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [9] NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [10] NIST SP 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [11] NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes using Discrete Logarithm Cryptography (Revised), <http://csrc.nist.gov/publications/PubsFIPS.html>
- [12] NIST SP 800-67 Revision 1, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [13] NIST SP 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, <http://csrc.nist.gov/publications/PubsFIPS.html>
- [14] RSA Laboratories, "PKCS #11 v2.20: Cryptographic Token Interface Standard", 2004.
- [15] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", CRYPTO '96, Lecture Notes In Computer Science, Vol. 1109, pp. 104-113, Springer-Verlag, 1996. <http://www.cryptography.com/timingattack/>
- [16] D. Boneh and D. Brumley, "Remote Timing Attacks are Practical", <http://crypto.stanford.edu/~dabo/abstracts/ssl-timing.html>
- [17] C. Percival, "Cache Missing for Fun and Profit", <http://www.daemonology.net/papers/htt.pdf>
- [18] N. Ferguson and B. Schneier, Practical Cryptography, Sec. 16.1.4 "Checking RSA Signatures", p. 286, Wiley Publishing, Inc., 2003.