FIPS 140-2 Non-Proprietary Security Policy

Oracle Linux 7 libgcrypt Cryptographic Module

FIPS 140-2 Level 1 Validation

Software Version: R7-4.0.0

Date: September 27th, 2021

Title: Oracle Linux 7 libgcrypt Cryptographic Module Security Policy

Date: September 27th, 2021

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

TABLE OF CONTENTS

List of Tables

List of Figures

1. Introduction

1.1 Overview

This document is the Security Policy for the Oracle Linux 7 libgcrypt Cryptographic Module by Oracle Corporation. Oracle Linux 7 libgcrypt Cryptographic Module is also referred to as “the Module or Module”. 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 libgcrypt Cryptographic Module functions in order to meet the FIPS 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 libgcrypt 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 libgcrypt 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 libgcrypt Cryptographic Module

2.1 Functional Overview

The Oracle Linux 7 libgcrypt Cryptographic Module is a software library implementing general purpose cryptographic algorithms. The module provides cryptographic services to applications running in the user space of the underlying operating system through an application program interface (API).

2.2 FIPS 140-2 Validation Scope

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

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

Table 1: FIPS 140-2 Security Requirements

3. Cryptographic Module Specification

3.1 Definition of the Cryptographic Module

The Oracle Linux 7 libgcrypt Cryptographic Module 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 file and its integrity check HMAC file, which are delivered through the Oracle Yum Server Package Manager (RPM) as listed below:

All components of the module will be in the libgcrypt RPM version 1.5.3-14.el7.x86_64. The following RPMs files are part of the module:

- libgcrypt-1.5.3-14.el7.x86_64

When installed on the system, the module comprises the following files:

- /usr/lib64/libgcrypt.so.11
- /usr/lib64/.libgcrypt.so.11.hmac

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

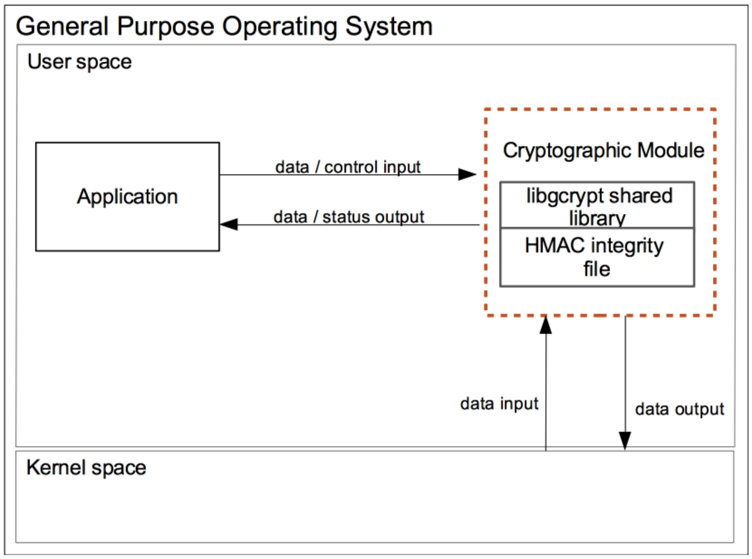

Figure 1: Oracle Linux 7 libgcrypt Logical Cryptographic Boundary

3.2 Definition of the Physical Cryptographic Boundary

The module is aimed to run on a general-purpose computer. No components are excluded from the requirements of FIPS PUB 140-2. The physical boundary is the surface of the case of the target platform, as shown in the diagram below:

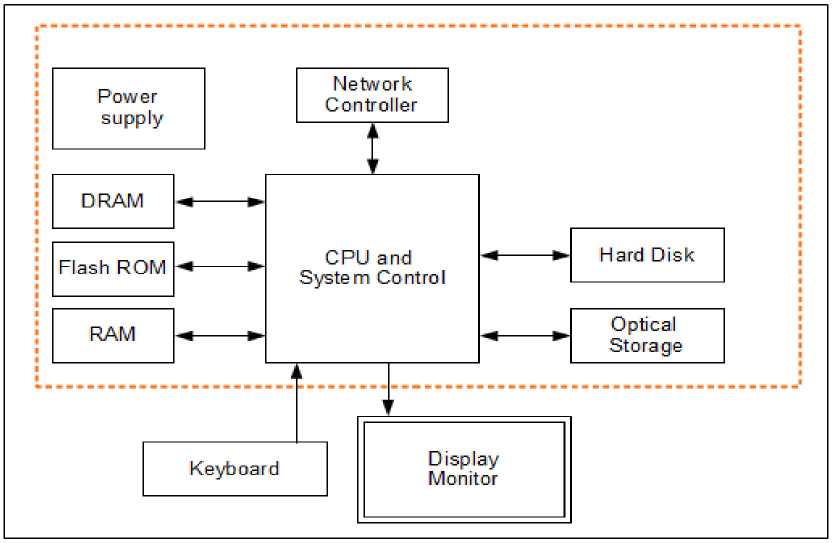

Figure 2: Oracle Linux 7 libgcrypt Hardware Block Diagram

3.3 Modes of Operation

The module supports two modes of operation: FIPS approved and non-approved modes. The mode is implicitly assumed depending on the services and security functions invoked. The module turns to the FIPS approved mode after the initialization and the power-on self-tests have completed successfully. Using a non-Approved service listed in Table 9 will result in the module implicitly entering the non-FIPS mode of operation

3.4 Approved Security Functions

The Oracle Linux 7 libgcrypt Cryptographic Module contains the following FIPS Approved Algorithms:

Approved Security Functions		Certificate
Symmetric Algorithms		
AES	**CBC, ECB, CFB128, OFB** (e/d; 128 , 192 , 256); **CTR** (ext only; 128 , 192 , 256)	C 709
Triple-DES (3-Key)	**TCBC** ; **TECB, TCFB64, TOFB** (KO 1 e/d,) ; **CTR** (ext only) Three-Key Triple-DES e/d; Two-key Triple-DES decryption only for legacy use.	C 709
Secure Hash Standard (SHS)		
SHS	**SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only)	C 709
Data Authentication Code		
HMAC	**HMAC-SHA1** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA224** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA256** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA384** (Key Size Ranges Tested: KS<BS KS=BS KS>BS) **HMAC-SHA512** (Key Size Ranges Tested: KS<BS KS=BS KS>BS)	C 709
Asymmetric Algorithms		

Approved Security Functions		Certificate
RSA	**FIPS186-4:** **186-4 KEY (gen):** FIPS186-4_fixed_e **PGM: (ProbRandom)** (2048 , 3072) **PPTT:** (C.2) **ALG[RSASSA-PKCS1_V1_5]** Sig (gen) (2048 SHA (224, 256, 384, 512)) (3072 SHA (224, 256, 384, 512)) Sig (ver) (1024 SHA (1, 224, 256, 384, 512)) (2048 SHA (1, 224, 256, 384, 512)) (3072 SHA (1, 224, 256, 384, 512)) **[RSASSA-PSS]:** Sig (Gen): (2048 SHA (224 SaltLen(20), 256 SaltLen(20), 384 SaltLen(20), 512 SaltLen(20))) (3072 SHA (224 SaltLen(20), 256 SaltLen(20), 384 SaltLen(20), 512 SaltLen(20))) Sig(Ver): (1024 SHA (1 SaltLen(20), 224 SaltLen(20), 256 SaltLen(20), 384 SaltLen(20), 512 SaltLen(20))) (2048 SHA (1 SaltLen(20), 224 SaltLen(20), 256 SaltLen(20), 384 SaltLen(20), 512 SaltLen(20))) (3072 SHA (1 SaltLen(20), 224 SaltLen(20), 256 SaltLen(20), 384 SaltLen(20), 512 SaltLen(20)))	C 709
DSA	**FIPS186-4:** **PQG(gen)PARMS TESTED:** [(2048, 224) SHA (224); (2048,256) SHA (256); (3072,256) SHA (256)] **KeyPairGen:** [(2048,224) ; (2048,256) ; (3072,256)] **SIG(gen)PARMS TESTED:** [(2048,224) SHA (224); (2048,256) SHA (256); (3072,256) SHA (256);] **SIG(ver)PARMS TESTED:** [(1024,160) SHA(1); (2048,224) SHA (224); (2048,256) SHA (256); (3072,256) SHA (256)]	C 709
Random Number Generation		
DRBG	**Hash_Based DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-256, SHA-384, SHA-512)] **HMAC_Based DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-256, SHA-384, SHA-512) **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128, AES-192, AES-256)]	C 709
CKG	In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP 800-133	Vendor Affirmed

Table 2: FIPS Approved Security Functions

3.5 Non-Approved but Allowed Security Functions

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

Algorithm	Usage
RSA key wrapping	key size between 2048 bits and 15360 bits or more; key establishment methodology provides between 112 and 256 bits of encryption strength.
NDRNG	Used for seeding SP 800-90A DRBG.

Table 3: Non-Approved but Allowed Security Functions

3.6 Non-Approved Security Functions

The following services are non-Approved and use of these algorithms will put the module in the non-approved mode of operation implicitly. The services associated with these algorithms are specified in section 7.3:

Algorithm	Usage
ARC4	Encrypt/Decrypt
Blowfish	Encrypt/Decrypt
Camellia	Encrypt/Decrypt
CAST5	Encrypt/Decrypt
DES	Encrypt/Decrypt
IDEA	Encrypt/Decrypt
RC2	Encrypt/Decrypt
SEED	Encrypt/Decrypt
Serpent	Encrypt/Decrypt
Twofish	Encrypt/Decrypt
2-Key Triple-DES	Encryption
RSA	Key generation, signature generation, key wrapping with keys less than 2048 bits
RSA	Signature verification with keys smaller than 1024 bits modulus size
DSA	Parameter verification, Parameter/Key generation/Signature generation with keys not listed in Table 3
GOST	28147 Encryption, R 34.11-94 Hashing, R 34.11.2012 (Stribog) Hashing
CSPRNG	Generating random numbers
Tiger	Hashing
MD4	Hashing
MD5	Hashing
Whirlpool	Hashing
RIPEMD 160	Hashing
El Gamal	Key generation/encryption/decryption/signature generation and verification
CRC32	Cyclic redundancy check
OpenPGP Salted and Iterated/Salted	Password based KDF (RFC 4880)
SHA-1	Used for signature Generation

Table 4: Non-Approved Functions

4. Module Ports and Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs. The logical interfaces are the application program interface (API) through which applications request services.

Table below shows a mapping of FIPS 140 interfaces to logical ports:

FIPS 140 Interface	Module Interfaces
Data Input	API input parameters for data
Data Output	API output parameters for data
Control Input	API function calls, API input parameters
Status Output	API return codes and error message

Table 5: Mapping of FIPS 140 Logical Interfaces to Logical Ports

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the API function calls and the input parameters used to control the behavior of the module. The Status Output interface includes the return values of the API functions and status sent through log messages.

5. Physical Security

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

6. Operational Environment

The module operates in a modifiable operational environment per FIPS 140-2 level 1 specifications.

6.1 Tested Environments

The Module was tested on the following environments:

Operating Environment	Processor	Hardware
Oracle Linux 7.6 64-bit	Intel® Xeon® Silver 4114	Oracle Server X7-2
Oracle Linux 7.6 64-bit	AMD® EPYC® 7551	Oracle Server X7-2

Table 6: Tested Operating Environments

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 7.8 64-bit	Intel® Xeon® Platinum 8167M	Oracle Server X7-2c
Oracle Linux 7.8 64-bit	AMD® EPYC® 7551	Oracle Server E1
Oracle Linux 7.8 64-bit	Ampere Altra A1	Oracle Server A1
Oracle Linux 7.8 64-bit	Intel® Xeon® Ice Lake-SP	Oracle Server X9
Oracle Linux 7.8 64-bit	Intel® Xeon® Silver 4114	Oracle Server X7-2
Oracle Linux 7.8 64-bit	AMD® EPYC® 7551	Oracle Server X7-2
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® 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® 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® 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® E7-2800 v2/E7-4800 v2, v3 & v4/E7-8800 v2 & v4	Cisco UCS C460 M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge FC630
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-4600 v3	Dell PowerEdge FC830
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3	Dell PowerEdge M630 Blade

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

Operating Environment	Processor	Hardware
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® 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® E5-2600 v3	Inspur Yingxin NF5180M4
Oracle Linux 7.6 64-bit	Intel® Xeon® E5-2600 v3 & v4	Inspur Yingxin NF5280M4
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® 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® E7-4800 v4/E7-8800 v4	NEC NX7700x/A4012L-2
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

Operating Environment	Processor	Hardware
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® 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	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® 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

Table 7: Vendor Affirmed Operating Environments

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

6.3 Policy

The operating system is restricted to a single operator (concurrent operators are explicitly excluded). The application that request cryptographic services is the single user of the module, even when the application is serving multiple clients. In operational mode, the ptrace(2) system call, the debugger (gdb(1)), and strace(1) shall be not 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 supports the following roles:

- **User Role**: performs all services, except module installation.
- **Crypto Officer Role**: performs module installation.

The User and Crypto Officer roles are implicitly assumed by the entity accessing the module services.

7.2 FIPS Approved or allowed services

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

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X		Symmetric Encryption/ Decryption	Encrypts or decrypts a block of data using Triple-DES and/or AES	AES, Triple-DES Key	R, W, X
X		Get Key Length	cipher_get_keylen() function	N/A	-
X		Get Block Length	cipher_get_blocksize() function	N/A	-
X		Check Availability of Algorithm	cipher_get_blocksize() function	N/A	-
X		Hash	Secure Hash Algorithm (SHS) hashes a block of data.	N/A	-
X		Keyed Hash (HMAC)	authenticate data using HMAC-SHA	HMAC Key	R, W, X
X		Digital Signature Generation and Verification	Sign and verify operation	RSA, DSA Keys	R, W, X
X		Asymmetric Key Generation	Key pair generation for RSA or DSA	RSA, DSA Keys	R, W, X
X		Asymmetric Encryption/ Decryption	Encrypts or decrypts using Approved RSA key size	RSA Key pair	R, W, X
X		Random Number generation	Generate random numbers based on the SP 800-90A DRBG	Entropy input string, seed and internal state	W, X
X		Self-Tests	Performs power-up self-test on demand	N/A	-

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X		Zeroization	Zeroization performed by functions gcry_cipher_close(), gcry_sexp_release(), gcry_drbg_uninstantiate(),gcry_md_close()	All CSP's listed above.	W
X		Show Status	Show status of the module state	None	-
	X	Module installation	Install and configure the module	None	-

R – Read, W – Write, X – Execute

Table 8: FIPS Approved or allowed Services and Descriptions

7.3 Non-FIPS Approved Services and Descriptions

The following table shows the services available in the non-FIPS mode:

U	CO	Service Name	Service Description	Keys	Access Type(s)
X		Symmetric Encryption/Decryption	Encrypts or decrypts using non-Approved algorithms listed in Table 4.	ARC4, Blowfish, Camellia, CAST5, DES, IDEA, RC2, GOST, SEED, Serpent, Twofish, 2-Key Triple-DES Key (encrypt only)	R, W, X
X		Asymmetric Encryption/Decryption	Encrypts or decrypts using non-Approved RSA key size or using ElGamal	RSA key < 2048 bits, ElGamal keys	R, W, X
X		Digital Signature Generation and Verification	Sign or verify operations with non-Approved RSA or DSA key or signature generation using SHA-1 or ElGamal keys	RSA, DSA keys listed in Table 4, ElGamal keys	R, W, X
X		Asymmetric Key Generation	Generation of non-Approved RSA and DSA keys or ElGamal keys	RSA, DSA keys listed in Table 4, ElGamal keys	R, W, X
X		Hash	Hashing using non-Approved hash functions listed in Table 4.	N/A	-
X		Keyed Hash (HMAC)	HMAC-SHA Service with Key Sizes < 112 bits	HMAC Key	R, W, X
X		Cyclic Redundancy code	CRC 32	N/A	-
X		Random Number generation	Generating Random Numbers using CSPRNG	Entropy input string, seed and internal state	R, W, X
X		Password based KDF	OpenPGP Salted and Iterated/Salted (RFC 4880)	Derived key	R, W, X

Table 9: Non-FIPS Approved Operator Services and Descriptions

7.4 Operator Authentication

The module is a Level 1 software-only cryptographic module and does not implement authentication. The role is implicitly assumed based on the service requested.

8. Cryptographic Key Management

Below is a list of the CSPs/keys details concerning storage, generation and zeroization:

CSP Name	Generation	Input	Storage	Zeroization
AES Keys	N/A.	The Key is passed into the module via API input parameter	RAM	Automatically zeroized when freeing the cipher handler by calling gcry_cipher_close ()
Triple-DES Keys			RAM	
DSA Private Key	Generated using FIPS 186-4 and the random value used is generated using SP800- 90A DRBG	The Key is passed into the module via API input parameter	RAM	Automatically zeroized when freeing the cipher handler by calling gcry_sexp_release ()
RSA Private Key			RAM	
DRBG Entropy Input String (Seed)	N/A	obtained from NDRNG	RAM	Automatically zeroized when freeing DRBG handler by calling gcry_drbg_uninstantiate()
DRBG internal state (V, C, Key)	Generated during DRBG initialization as defined in SP 800-90A	N/A	RAM	
HMAC Key	N/A.	The Key is passed into the module via API input parameter.	RAM	Automatically zeroized when freeing the cipher handler by calling gcry_md_close()

Table 10: CSP Table

8.1 Random Number Generation

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90A] for the creation of key components of asymmetric keys and for random number generation. The DRBG is initialized during module initialization. The module by default loads the HMAC_DRBG with SHA-256 which is used internally for the module's key generation function.

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

The module performs continuous tests on the output of the DRBG to ensure that consecutive random numbers do not repeat. The underlying operating system performs the continuous test on the NDRNG which is outside the module's logical boundary but inside the physical boundary.

8.2 Key Generation

The Key Generation methods implemented in the module for Approved services in FIPS mode is compliant with [SP 800-133].

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

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

8.3 Key/CSP Storage

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

8.4 Key/CSP Zeroization

The memory occupied by keys is allocated by regular memory allocation operating system calls. The application is responsible for calling the appropriate destruction functions listed in Table 10. The destruction functions overwrite the memory occupied by keys with “zeros” and deallocates the memory with the regular memory deallocation operating system call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process.

8.5 Key Establishment

The module provides RSA key wrapping which is an allowed method of key transport according to IG D.9. The security strength for RSA key wrapping is listed in Table 3.

9. Self-Tests

The Module perform self-tests to ensure the integrity of the Module, and the correctness of the cryptographic functionality at start up. In addition, some functions require continuous verification, such as the Random Number Generator. All of these tests are listed and described in this section.

9.1 Power-Up Self-Tests

The module performs power-on self-tests (POST) automatically during loading of the module by making use of default entry point (DEP) without any user intervention. While the power-up tests are in progress, services are not available and input or output is not possible: the module is single-threaded and will not return to the calling application until the self-tests are completed successfully. If any of the self-test fails module enters Error state. Only if all the self-tests are successful then the module is operational and cryptographic functionality is available for use.

The integrity of the module is verified by comparing the HMAC-SHA-256 value calculated at run time with the HMAC value stored in the hmac file that was computed at build time.

Algorithm	Test
AES (128, 192, 256)	KATs for encryption and decryption are tested separately
Triple-DES	KAT for encryption and decryption are tested separately
DSA	PCT for signature generation and verification
RSA	KAT for signature generation and verification
SHA	KAT for (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
HMAC	KAT for (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512)
DRBG	KAT for (Hash, HMAC, and CTR)
Module Integrity	HMAC-SHA-256 integrity test

Table 11: Power-On Self-Tests

9.2 On Demand Self-Tests

The module provides the Self-Test service to perform self-tests on demand. This service performs the same cryptographic algorithm tests executed during power-up. To invoke the on-demand self-tests, the user can make a call to the gcry_control (GCRYCTL_SELFTEST) function. During the execution of the on-demand self-tests, services are not available and no data output or input is possible.

9.3 Conditional Self-Tests

The following table provides the lists of the conditional self-tests. If any of the conditional test fails, the Module enters the Error state.

Algorithm	Test
DRBG	Continuous Random Number Generator Test
RSA key generation	Pairwise Consistency Test: signature generation and verification
DSA key generation	Pairwise Consistency Test: signature generation and verification
DRBG	SP 800-90A Section 11.3 Health Tests

Table 12: Conditional Self-Tests

9.4 DRBG Health Tests

The module supports the following health tests for the DRBG:

Algorithm	Test
DRBG Health Tests	Instantiate Function
	Generate Function
	Reseed Function

Table 13: DRBG Health Tests

9.5 Error State

The Module enters Error state with error message, on failure of POST or conditional test. In Error state, all data output is inhibited and no cryptographic operation is allowed. The error can be recovered by calling gcry_control(GCRYCTL_SELFTEST) function that reruns the POST. The module enters Fatal Error state when random numbers are requested in error state or when requesting cipher operations on deallocated handle. In Fatal Error state the module is aborted and is not available for use. The module needs to be reloaded in order to recover from Fatal Error state.

10. Guidance

10.1 Crypto-Officer Guidance

The version of the RPM containing the FIPS validated Module is stated in section 3.1 above. For proper operation of the in-module integrity verification, the CO shall ensure prelinking is disabled on all system files. This can be done by setting PRELINKING=no in the /etc/sysconfig/prelink configuration file. If the libraries were already prelinked, the prelink should be undone on all the system files using the 'prelink -u -a' command.

To configure the operating environment to support FIPS 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. Recreate the INITRAMFS image:

```
# dracut -f
```

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

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

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

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

# blkid /dev/sda1

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

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

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

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

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

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

iii. Save the changes.

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

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

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

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

5. 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, the CO shall ensure prelinking is 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 can be performed 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
   ```

6. Reboot the system

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

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

   The response should be “1”

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

10.2 User Guidance

Applications using libgcrypt need to call gcry_control(GCRYCTL_INITIALIZATION_FINISHED, 0) after initialization is done: that ensures that the DRBG is properly seeded, among others. gcry_control(GCRYCTL_TERM_SECMEM) needs to be called before the process is terminated. The function gcry_set_allocation_handler() may not be used.

The user must not call malloc/free to create/release space for keys, let libgcrypt manage space for keys, which will ensure that the key memory is overwritten before it is released. See the documentation file doc/gcrypt.texi within the source code tree for complete instructions for use.

The information pages are included within the developer package. The user can find the documentation at the following location after having installed the developer package:

/usr/share/info/gcrypt.info-1.gz
/usr/share/info/gcrypt.info-2.gz
/usr/share/info/gcrypt.info.gz

10.2.1 Triple-DES Keys

Data encryption using the same three-key Triple-DES key shall not exceed 2^{16} Triple-DES (64-bit) blocks, in accordance to [SP800-67] and IG A.13 in [FIPS140-2-IG]. The user of the module is responsible for ensuring the module's compliance with this requirement.

11. Mitigation of Other Attacks

libgcrypt uses a blinding technique for RSA decryption to mitigate real world timing attacks over a network: Instead of using the RSA decryption directly, a blinded value (y = x·re mod n) is decrypted and the unblinded value (x' = y'·r-1 mod n) returned. The blinding value “r” is a random value with the size of the modulus “n” and generated with `GCRY_WEAK_RANDOM' random level.

Weak Triple-DES keys are detected as follows:

In DES there are 64 known keys which are weak because they produce only one, two, or four different subkeys in the subkey scheduling process. The keys in this table have all their parity bits cleared.

```
static byte weak_keys[64][8] =
{
        { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }, /*w weak keys*/
        { 0x00, 0x00, 0x1e, 0x1e, 0x00, 0x00, 0x0e, 0x0e },
        { 0x00, 0x00, 0xe0, 0xe0, 0x00, 0x00, 0xf0, 0xf0 },
        { 0x00, 0x00, 0xfe, 0xfe, 0x00, 0x00, 0xfe, 0xfe },
        { 0x00, 0x1e, 0x00, 0x1e, 0x00, 0x0e, 0x00, 0x0e }, /*sw semi-weak keys*/
        { 0x00, 0x1e, 0x1e, 0x00, 0x00, 0x0e, 0x0e, 0x00 },
        { 0x00, 0x1e, 0xe0, 0xfe, 0x00, 0x0e, 0xf0, 0xfe },
        { 0x00, 0x1e, 0xfe, 0xe0, 0x00, 0x0e, 0xfe, 0xf0 },
        { 0x00, 0xe0, 0x00, 0xe0, 0x00, 0xf0, 0x00, 0xf0 }, /*sw*/
        { 0x00, 0xe0, 0x1e, 0xfe, 0x00, 0xf0, 0x0e, 0xfe },
        { 0x00, 0xe0, 0xe0, 0x00, 0x00, 0xf0, 0xf0, 0x00 },
        { 0x00, 0xe0, 0xfe, 0x1e, 0x00, 0xf0, 0xfe, 0x0e },
        { 0x00, 0xfe, 0x00, 0xfe, 0x00, 0xfe, 0x00, 0xfe }, /*sw*/
        { 0x00, 0xfe, 0x1e, 0xe0, 0x00, 0xfe, 0x0e, 0xf0 },
        { 0x00, 0xfe, 0xe0, 0x1e, 0x00, 0xfe, 0xf0, 0x0e },
        { 0x00, 0xfe, 0xfe, 0x00, 0x00, 0xfe, 0xfe, 0x00 },
        { 0x1e, 0x00, 0x00, 0x1e, 0x0e, 0x00, 0x00, 0x0e },
        { 0x1e, 0x00, 0x1e, 0x00, 0x0e, 0x00, 0x0e, 0x00 }, /*sw*/
        { 0x1e, 0x00, 0xe0, 0xfe, 0x0e, 0x00, 0xf0, 0xfe },
        { 0x1e, 0x00, 0xfe, 0xe0, 0x0e, 0x00, 0xfe, 0xf0 },
        { 0x1e, 0x1e, 0x00, 0x00, 0x0e, 0x0e, 0x00, 0x00 },
        { 0x1e, 0x1e, 0x1e, 0x1e, 0x0e, 0x0e, 0x0e, 0x0e }, /*w*/
        { 0x1e, 0x1e, 0xe0, 0xe0, 0x0e, 0x0e, 0xf0, 0xf0 },
        { 0x1e, 0x1e, 0xfe, 0xfe, 0x0e, 0x0e, 0xfe, 0xfe },
        { 0x1e, 0xe0, 0x00, 0xfe, 0x0e, 0xf0, 0x00, 0xfe },
        { 0x1e, 0xe0, 0x1e, 0xe0, 0x0e, 0xf0, 0x0e, 0xf0 }, /*sw*/
        { 0x1e, 0xe0, 0xe0, 0x1e, 0x0e, 0xf0, 0xf0, 0x0e },
        { 0x1e, 0xe0, 0xfe, 0x00, 0x0e, 0xf0, 0xfe, 0x00 },
        { 0x1e, 0xfe, 0x00, 0xe0, 0x0e, 0xfe, 0x00, 0xf0 },
        { 0x1e, 0xfe, 0x1e, 0xfe, 0x0e, 0xfe, 0x0e, 0xfe }, /*sw*/
        { 0x1e, 0xfe, 0xe0, 0x00, 0x0e, 0xfe, 0xf0, 0x00 },
        { 0x1e, 0xfe, 0xfe, 0x1e, 0x0e, 0xfe, 0xfe, 0x0e },
        { 0xe0, 0x00, 0x00, 0xe0, 0xf0, 0x00, 0x00, 0xf0 },
        { 0xe0, 0x00, 0x1e, 0xfe, 0xf0, 0x00, 0x0e, 0xfe },
        { 0xe0, 0x00, 0xe0, 0x00, 0xf0, 0x00, 0xf0, 0x00 }, /*sw*/
        { 0xe0, 0x00, 0xfe, 0x1e, 0xf0, 0x00, 0xfe, 0x0e },
        { 0xe0, 0x1e, 0x00, 0xfe, 0xf0, 0x0e, 0x00, 0xfe },
        { 0xe0, 0x1e, 0x1e, 0xe0, 0xf0, 0x0e, 0x0e, 0xf0 },
        { 0xe0, 0x1e, 0xe0, 0x1e, 0xf0, 0x0e, 0xf0, 0x0e }, /*sw*/
        { 0xe0, 0x1e, 0xfe, 0x00, 0xf0, 0x0e, 0xfe, 0x00 },
        { 0xe0, 0xe0, 0x00, 0x00, 0xf0, 0xf0, 0x00, 0x00 },
        { 0xe0, 0xe0, 0x1e, 0x1e, 0xf0, 0xf0, 0x0e, 0x0e },
        { 0xe0, 0xe0, 0xe0, 0xe0, 0xf0, 0xf0, 0xf0, 0xf0 }, /*w*/
        { 0xe0, 0xe0, 0xfe, 0xfe, 0xf0, 0xf0, 0xfe, 0xfe },
        { 0xe0, 0xfe, 0x00, 0x1e, 0xf0, 0xfe, 0x00, 0x0e },
        { 0xe0, 0xfe, 0x1e, 0x00, 0xf0, 0xfe, 0x0e, 0x00 },
        { 0xe0, 0xfe, 0xe0, 0xfe, 0xf0, 0xfe, 0xf0, 0xfe }, /*sw*/
        { 0xe0, 0xfe, 0xfe, 0xe0, 0xf0, 0xfe, 0xfe, 0xf0 },
        { 0xfe, 0x00, 0x00, 0xfe, 0xfe, 0x00, 0x00, 0xfe },
        { 0xfe, 0x00, 0x1e, 0xe0, 0xfe, 0x00, 0x0e, 0xf0 },
        { 0xfe, 0x00, 0xe0, 0x1e, 0xfe, 0x00, 0xf0, 0x0e },
        { 0xfe, 0x00, 0xfe, 0x00, 0xfe, 0x00, 0xfe, 0x00 }, /*sw*/
        { 0xfe, 0x1e, 0x00, 0xe0, 0xfe, 0x0e, 0x00, 0xf0 },
        { 0xfe, 0x1e, 0x1e, 0xfe, 0xfe, 0x0e, 0x0e, 0xfe },
        { 0xfe, 0x1e, 0xe0, 0x00, 0xfe, 0x0e, 0xf0, 0x00 },
        { 0xfe, 0x1e, 0xfe, 0x1e, 0xfe, 0x0e, 0xfe, 0x0e }, /*sw*/
        { 0xfe, 0xe0, 0x00, 0x1e, 0xfe, 0xf0, 0x00, 0x0e },
        { 0xfe, 0xe0, 0x1e, 0x00, 0xfe, 0xf0, 0x0e, 0x00 },
        { 0xfe, 0xe0, 0xe0, 0xfe, 0xfe, 0xf0, 0xf0, 0xfe },
        { 0xfe, 0xe0, 0xfe, 0xe0, 0xfe, 0xf0, 0xfe, 0xf0 }, /*sw*/
        { 0xfe, 0xfe, 0x00, 0x00, 0xfe, 0xfe, 0x00, 0x00 },
        { 0xfe, 0xfe, 0x1e, 0x1e, 0xfe, 0xfe, 0x0e, 0x0e },
        { 0xfe, 0xfe, 0xe0, 0xe0, 0xfe, 0xfe, 0xf0, 0xf0 },
        { 0xfe, 0xfe, 0xfe, 0xfe, 0xfe, 0xfe, 0xfe, 0xfe } /*w*/ };
```

Acronyms, Terms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CCCS	Canadian Centre for Cyber Security
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standard
FSM	Finite State Model
HMAC	(Keyed) Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
KDF	Key Derivation Function
NIST	National Institute of Standards and Technology
NDRNG	Non-Deterministic Random Number Generator
OFB	Output Feedback Mode
OS	Operating System
PBKDF	Password Based Key Derivation Function
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptographic Standard
POST	Power On Self-Test
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
PUB	Publication
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
TDES	Triple Data Encryption Standard

Table 14: Acronyms

References

Reference	Full Specification Name
[FIPS 140-2]	Security Requirements for Cryptographic modules, May 25, 2001
[FIPS 180-4]	Secure Hash Standard (SHS) March 2012 http://csrc.nist.gov/publications/fips/fips180-4/fips 180-4.pdf
[FIPS 186-4]	Digital Signature Standard (DSS) July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
[FIPS 197]	Advanced Encryption Standard November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[FIPS 198-1]	The Keyed Hash Message Authentication Code (HMAC) July 2008 http://csrc.nist.gov/publications/fips/fips198 1/FIPS-198 1 final.pdf
[PKCS #1]	Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
[RFC 5649]	Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm September 2009 http://www.ietf.org/rfc/rfc5649.txt
[SP 800-38A]	NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf
[SP 800-38B]	NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication June 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf
[SP 800-38C]	NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality May 2004 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
[SP 800- 67R1]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-89]	Recommendation for Obtaining Assurances for Digital Signature Applications
[SP 800-90]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[SP 800- 131Ar2]	Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

Table 15: References