Red Hat Enterprise Linux 6.2 dm-crypt Cryptographic Module v2.0

FIPS 140-2 Security Policy

Version 1.4

Last Update: 2013-04-03

Contents

1 Cryptographic Module Specification

This document is the non-proprietary security policy for the Red Hat Enterprise Linux 6.2 dm-crypt Cryptographic Module, and was prepared as part of the requirements for conformance to Federal Information Processing Standard (FIPS) 140-2, Level 1.

The following section describes the module and how it complies with the FIPS 140-2 standard in each of the required areas.

1.1 Description of Module

The Red Hat Enterprise Linux 6.2 dm-crypt Cryptographic Module is a software only cryptographic module that provides disk management and transparent partial or full disk encryption. Partial disk encryption encrypts only one or more partitions, leaving at least one partition as plaintext. Full disk encryption ensures that the entire disk is encrypted, including any Swap partition, but excluding /boot.

The following table shows the overview of the security level for each of the eleven sections of the validation.

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

Table 1: Security Levels

The module has been tested on the following multi-chip standalone platforms:

Manufacturer	Model	O/S & Ver.
HP	Proliant DL585	Red Hat Enterprise Linux 6.2 (In Single User Mode)
IBM	HS22	Red Hat Enterprise Linux 6.2 (In Single User Mode)

Table 2: Tested Platforms

The files that make up the module are:

- the contents of cryptsetup-luks-libs RPM package (version 1.2.0-6.el6)
- the contents of the fipscheck RPM package (version 1.2.0-7.el6)

- the contents of the fipscheck-lib RPM package (version 1.2.0-7.el6)
- kernel module /lib/modules/$(uname -r)/kernel/drivers/md/dm-crypt.ko
- the dracut-fips package with the version of the RPM file of 004-284.el6_3.1. This RPM version is provided with RHBA:2012-1318 accessible on the Red Hat Network.
- the OpenSSL FIPS 140-2 module with the certificate #1758 (bound module)
- the NSS FIPS 140-2 module with the certificate #1837 (bound module)
- the libgcrypt FIPS 140-2 module defined with the certificate #1757 (bound module)
- the kernel crypto API FIPS 140-2 module defined with the certificate #1901 (bound module)

The installer uses LVM (logical volume manager) to provide transparent storage space management. LVM is implemented using device mapper enabling setup of the following basic configurations:

1. disk | dm-crypt | <- (application)
2. disk | LVM PV | LVM LVs | dm-crypt | <- (application)
3. disk | dm-crypt | LVM PV | LVM LVs| <- (application)

In configuration 1, the installer supports encryption on a disk partition without using LVM.

In configuration 2, the installer supports encryption on a LV (logical volume) in LVM. In this case, only part of the LVM is encrypted. For example, the root partition is unencrypted, but the homes volume is encrypted.

In configuration 3, the installer supports encryption on a disk partition, which is a PV (physical volume) for LVM. In this case, the encryption is under LVM, everything is encrypted and LVM just allocates space over some encrypted device.

The module is a FIPS 140-2 security level 1 multichip standalone module. The Linux platform is configured in single user mode.

1.2 Description of Modes of Operations

The module must always be configured as outlined in section 9.1. The module can be operated in FIPS mode or non-FIPS mode depending on what cryptographic functions are invoked.

The table outlines the approved and non-approved functions provided by the specified components (for specific algorithm modes supported please see the Security Policy for the external FIPS 140-2 modules referenced).

Component	Approved functions	Non-approved functions
Kernel Crypto API Cryptographic Module performing the encryption and decryption of file system data and integrity check. (FIPS 140-2 Cert# 1901)	AES encrypt / decrypt 128,192, 256 bit (Certs. #1968, #1969, #1970, #1971 and #1972) SHA1, SHA224, SHA384, SHA512 (Certs. #1725 and #1726) Triple-DES encrypt / decrypt (Certs #1278 and #1279) DSA (Certs #628 and #629)	CTR mode (internal counter source) CBC encryption mode without the use of ESSIV IV handling mechanism DES XTS using AES192; although technically usable, CAVS currently does not provide a means to test it
Libgcrypt Cryptographic Module supporting the libcryptsetup LUKS volume key management, key generation, and integrity check. (FIPS 140-2 Cert# 1757)	HMAC-SHA-1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 (Certs. #1128, #1131, #1132 and #1133) SHA1, SHA224, SHA256, SHA384, SHA512 (Certs. #1657, #1660, #1661 and #1662) RNG (X9.31) (Certs. #988, #991, #992 and #993)	N/A
OpenSSL Cryptographic Module supporting the integrity check of libcryptsetup. (FIPS 140-2 Cert# 1758)	HMAC-SHA256 (Certs. #1129, #1130, #1134 and #1135) SHA256 (Certs. #1658, #1659, #1663 and #1664)	N/A
NSS Cryptographic Module used by sha512hmac performing its own integrity check and the integrity check of the static kernel binary. (FIPS 140-2 Cert# 1837)	HMAC-SHA512 (Certs. #1199 and #1200) DSA (Certs. #634 and #635) SHA512 (Certs. #1741 and #1742)	N/A
libcryptsetup performing the volume key management operations using the LUKS format	PBKDF as specified in SP800-132 section 5.3 protecting the volume key according to SP800-132 section 5.4 Option 2a	basic (plain) dm-crypt mappings without using the LUKS extensions

The non-approved functions shall not be used in FIPS approved mode. The following list references the documents where the list of all non-approved algorithms beyond those callable via the Red Hat Enterprise Linux 6.2 dm-crypt Cryptographic Module are specified:

- Kernel Crypto API: see Security Policy of Kernel Crypto API

- Ligcrypt: see API documentation provided with libgcrypt
- OpenSSL: see man page API documentation provided with libgcrypt
- NSS: see Security Policy of NSS

1.3 Cryptographic Module Boundary

The Red Hat Enterprise Linux 6.2 dm-crypt Cryptographic Module physical boundary is defined by the surface of the case of the test platform. The logical module boundary is depicted in the software block diagram.

1.3.1 Hardware Block Diagram

This figure illustrates the various data, status and control paths through the cryptographic module. The physical boundary consists of the opaque enclosure surrounding the COTS PC system. The module consists of standard integrated circuits, including processors, and memory. The module does not include any security-relevant, semi- or custom integrated circuits or other active electronic circuit elements. The physical module includes power inputs and outputs, and internal power supplies. The cryptographic boundary contains only the security-relevant software elements that comprise the module.

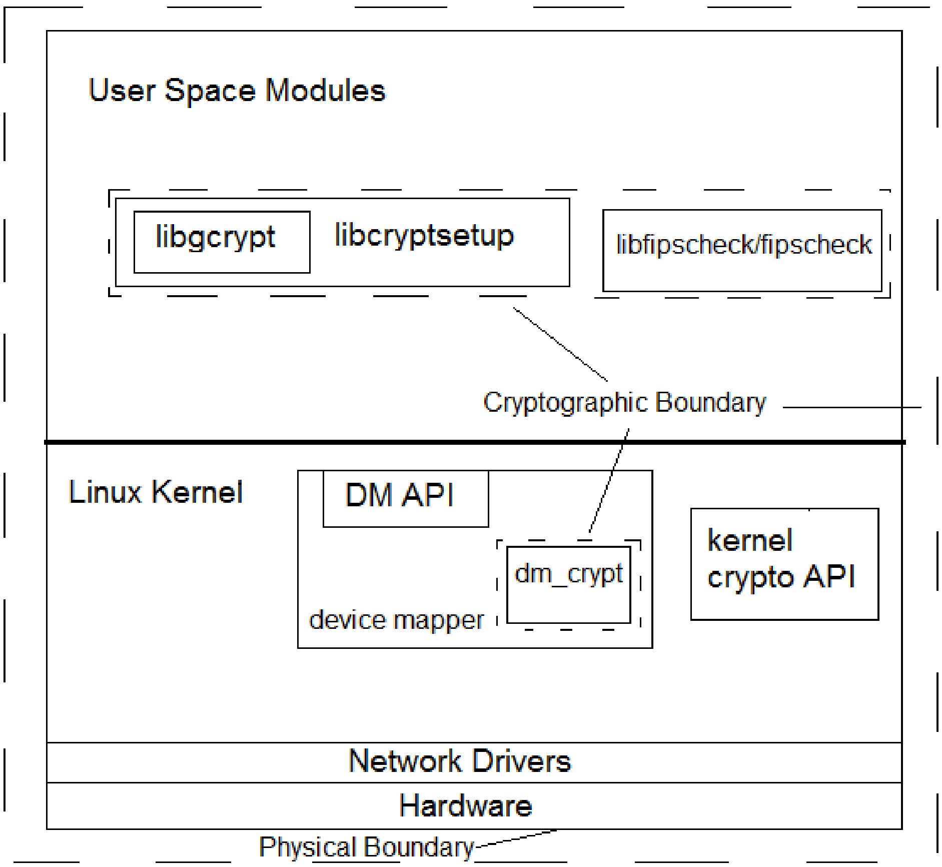

Figure 1: Hardware Block Diagram

1.3.2 Software Block Diagram

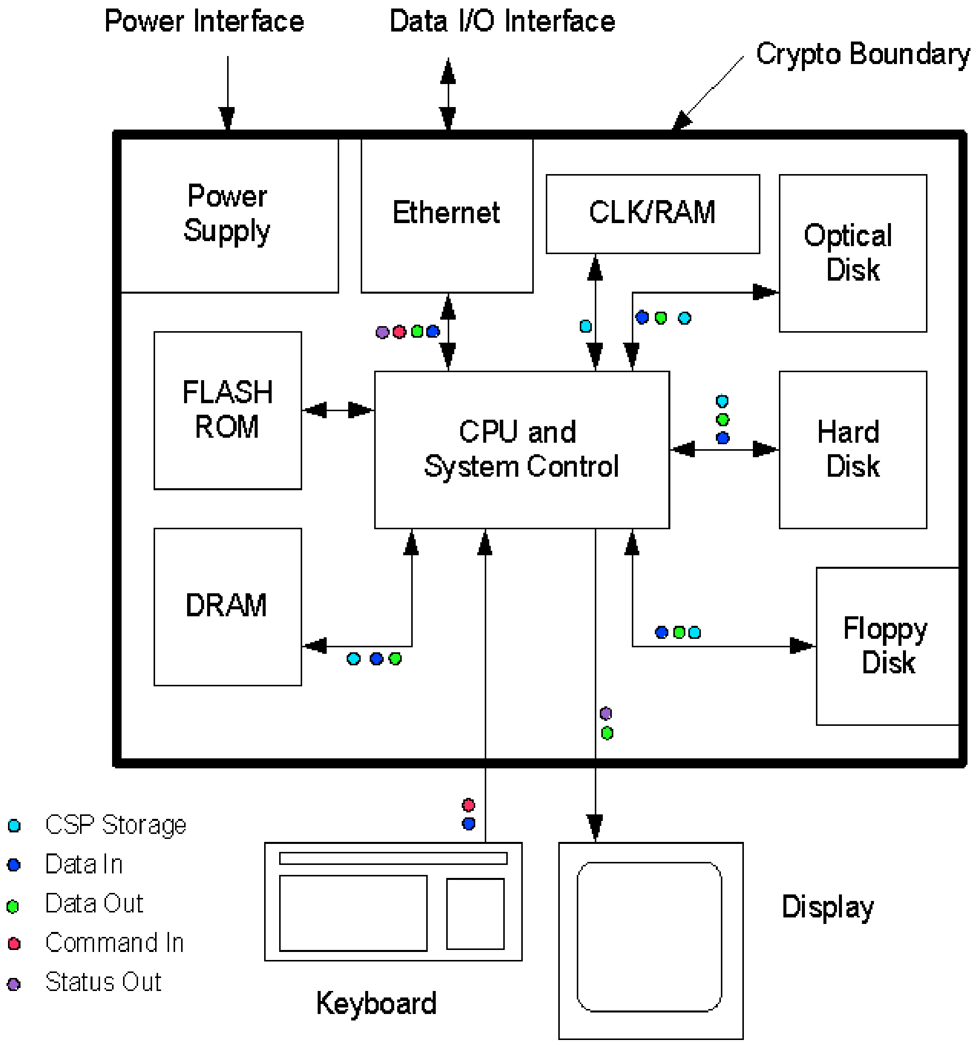

Figure 2:Software Block Diagram

1.4 Red Hat Enterprise Linux 6.2 Cryptographic Modules and FIPS 140-2 Certification

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

The certification is performed at FIPS 140-2 level 1, a software only certification that does not make any claims about the hardware enclosure. This allows vendors to develop their own higher level FIPS 140-2-certified modules using cryptographic modules.

The following cryptographic modules are included in the RHEL6.2 certification:

1 Kernel Crypto API - a software only cryptographic module that provides general-purpose cryptographic services to the remainder of the Linux kernel
2 Disk Volume Encryption - provides disk management and transparent partial or full disk encryption; Partial disk encryption encrypts only one or more partitions, leaving at least one partition as plaintext.
3 Libgcrypt- supplies general cryptographic support for the Red Hat Enterprise Linux user space
4 OpenSSL - a software library supporting cryptographic algorithms for general use by vendors
5 OpenSSH-Server - supplies cryptographic support for the SSH protocol
6 OpenSSH-Client - supplies cryptographic support for the SSH protocol
7 Openswan - provides the IKE protocol version 1 key agreement services required for IPSec

1.4.1 Platforms

The certification was performed on a 64-bit system, which are capable of executing both 32 and 64-bit code concurrently. Vendors can "transfer" the FIPS 140-2 certificate to the other similar platforms, provided the binary is only recompiled without changing the code. This is called a vendor assertion.

1.4.2 FIPS Approved Mode

Any vendor-provided FIPS certified cryptographic modules and services that use the RHEL6.2 underlying services to provide cryptographic functionality must use the RHEL6.2 services in their approved mode. Only operating in the approved mode ensures that FIPS 140-2 required self tests are executed and that ciphers are restricted to those that have been FIPS 140-2 certified by independent testing.

This section is given as guidance to developers to ensure the use cases are conformant with the initialization of the module. Although the modules always operate in FIPS mode when following the guidance, the initialization of the FIPS 140-2 module may consist of several phases which may be preempted by developers, but not administrators or users. The following explanation describes the initialization phase to ensure the FIPS 140-2 module initializes correctly.

If dm-crypt is used to encrypt a disk or partition, particularly when other modules may store cryptographic keys or other CSPs (critical security parameters) there, it must be in FIPS approved mode as described in dm-crypt Security Policy.

Since the kernel is in the approved mode, the remaining RHEL6.2 FIPS services (Crypto API, OpenSSL, Openswan, OpenSSH) start up in the approved mode by default. (Note that Openswan uses NSS for its cryptographic operations and NSS must explicitly be put into the approved mode with the modutil command.)

The approved mode for a module becomes effective as soon as the module power on self tests complete successfully and the module loads into memory. Self tests and integrity tests triggered in RHEL6.2 at startup or on the first invocation of the crypto module:

- kernel: during boot, before dm-crypt becomes available via dracut
- user space: before the user space module is available to the caller (i.e. for libraries during the initialization call, for applications: at load time)

See each module security policy for descriptions of self tests performed by each module and descriptions of how self test errors are reported for each module.

2 Cryptographic Module Ports and Interfaces

Function	Port
Command In	Libcryptsetup API, /proc/sys/crypto/fips_enabled, kernel command line option of “fips”
Status Out	Libcryptsetup API
Data In	Kernel system calls for file system write-like requests for devices protected by dm-crypt.
Data Out	Kernel system calls for file system read-like requests for devices protected by dm-crypt.

Table 3: Ports and Interfaces

3 Roles, Services and Authentication

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

3.1 Roles

Role	Services
User	• Kernel system calls on file system objects stored on devices protected by dm-crypt
Crypto Officer	• Installation • Generation and protection of volume key • Setup dm-crypt mapping, obtaining and insertion of volume key into kernel • Protect volume key with one or more passphrases • Self Tests • Show Status • Deconfigure dm-crypt mapping and zeroization of volume key

Table 4: Roles

3.2 Services

Table 5 shows services that use or affect stored cryptographic keys or CSPs. For each service, the key or CSP is indicated along with the type of access. This table only covers the approved services.

R - The item is read or referenced by the service

W - The item is written or updated by the service

Z - The item is zeroized by the service

Service	Category	Function	Role	Cryptographic Keys and CSPs Accessed	Access Type (RWZ)
Kernel system calls on file system objects stored on devices protected by dm-crypt	Symmetric Cipher, hash	Kernel system calls operating on file system objects	User, crypto officer	Volume key	R
Generation and protection of volume key	RNG PBKDF	Libcryptsetup API call	Crypto officer	Passphrase Volume key	RZ RWZ
Setup dm-crypt mapping, obtaining and insertion of volume key into	PBKDF	Libcryptsetup API call	Crypto officer	Passphrase Volume key	RZ RZ

Service	Category	Function	Role	Cryptographic Keys and CSPs Accessed	Access Type (RWZ)
kernel					
Protect volume key with one or more passphrases	PBKDF	Libcryptsetup API call	Crypto officer	Passphrase Volume key	RZ RWZ
Self Tests	Self Test (includes Integrity and known answer tests)	Invoked by restarting the module	Crypto officer	Software integrity Key	R
Show Status	Status	Libcryptsetup API call	Crypto officer	None	N/A
Deconfigure dm-crypt mapping and zeroization of volume key	None	Libcryptsetup API call	Crypto officer	Volume key	Z
Installation	None	None	Crypto officer	None	N/A

Table 5: Services

For additional services provided by the cryptographic module components that are FIPS 140-2 modules in their own right, please see the respective Security Policies.

3.3 Operator Authentication

There is no operator authentication. The assumption of a role is implicit by the action taken.

3.4 Mechanism and Strength of Authentication

At security level 1, authentication is not required.

4 Physical Security

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

5 Operational Environment

5.1 Applicability

This module will operate in a modifiable operational environment per the FIPS 140-2 specifications.

5.2 Policy

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

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

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

6 Cryptographic Key Management

The following table identifies the Cryptographic Keys and Critical Security Parameters (CSPs) used within the module. Cryptographic keys and CSPs are never output from the module in plaintext. An Approved key generation method is used to generate keys that are generated on the module.

6.1 Key Life Cycle Table

Key	Type	Generation or Source of Input	Establishment	Access by Role/Service	Entry and Output Method	Storage	Zeroization
Server Side Cryptographic Keys and CSPs							
Volume Key	AES 128, 192, or 256-bits TDES 168 bits	libgcrypt RNG	N/A	Data encryption - decryption	N/A	Stored in encrypted form (PBKDF)	Zeroized on a zeroize command
Software Integrity Key	DSA 2048-bit HMAC SHA-256 HMAC SHA-512	Generated off the module	N/A	Software integrity test (POST)	N/A	Stored in plaintext form	Zeroized on a zeroize command

Table 7: Cryptographic Keys and CSPs

Notes: The module ships without containing any keys and CSPs. When the module is configured, the crypto officer generates a volume key per dm-crypt mapping. Each volume key is stored in the first blocks of the disk device hosting the respective dm-crypt mapping.

6.2 Key Zeroization

For volatile memory, memset is included in deallocation operations. There are no restrictions when zeroizing any cryptographic keys and CSPs.

6.3 Random Number Generation

Libcryptsetup uses the random number generator provided by libgcrypt for generating the volume key.

Please see the Security Policy of libgcrypt for details on the random number generation and the seeding of of the random number.

7 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

Product Name and Model: HP ProLiant Server DL585 Series
Regulatory Model Number: HSTNS-1025
Product Options: All
EMC: Class A

Product Name and Model: IBM BladeCenter HS22 Series
Regulatory Model Number: 09-EMCRTP-0008
Product Options: All
EMC: Class A

8 Self Tests

8.1 Power-Up Tests

The module performs a power-on self test (POST) consisting of the software integrity test (using fipscheck) to verify the integrity of libcryptsetup. If the module fails the integrity test the module halts operation.

Cryptographic algorithms and critical security functions are provided by the cryptographic components constituting FIPS 140-2 modules in their own right that are listed in section 1.1. All algorithms and critical security functions are tested during initialization of those modules.

8.1.1 Software Integrity Test

The integrity check is implemented separately for validating the kernel space and the user space components.

The integrity check of all components provided with the kernel is described in the kernel crypto API Security Policy.

The integrity check of the user space components of the module is performed by the application fipscheck invoked during the startup of libcryptsetup when requesting cryptographic operations.

When the module starts, it exercises the software integrity test.

The user space integrity verification is performed as follows:

1. The libcryptsetup library links with the library libfipscheck.so which is intended to execute fipscheck to verify the integrity of the calling library file using HMAC SHA-256. Upon calling the FIPSCHECK_verify() function provided with libfipscheck.so, the fipscheck application is loaded and executed.
2. fipscheck loads and initializes the OpenSSL library which performs the integrity check of the OpenSSL library files using HMAC SHA-256.
3. The application fipscheck performs the integrity check of its application file as well as the libfipscheck.so library file using HMAC SHA-256 with the cipher provided by OpenSSL.
4. The fipscheck application performs the integrity check of the calling library. The fipscheck computes the HMAC SHA-256 checksum of the file and compares the computed value to the value stored inside the /path/to/library/.<libraryfilename>.hmac checksum file. The fipscheck application returns the appropriate exit value based on the comparison result (zero if the checksum is OK – which is enforced by the libfipscheck.so library).

9 Guidance

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

9.1 Cryptographic Officer Guidance

The version of the RPM containing the validated module is stated in section 1.1 above. The integrity of the RPM is automatically verified during the installation and the crypto officer shall not install the RPM file if the RPM tool indicates an integrity error.

The RPM package of the module can be installed by standard tools recommended for the installation of RPM packages on a Red Hat Enterprise Linux system (for example, yum, rpm, and the RHN remote management tool).

For proper operation of the in-module integrity verification, the prelink has to be disabled. This can be done by setting PRELINKING=no in the /etc/sysconfig/prelink configuration file. Existing prelinking, if any, should be undone on all the system files using the 'prelink -u -a' command.

To bring the module into FIPS approved mode, perform the following:

Install the dracut-fips package:

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

Recreate the INITRAMFS image:

```
# dracut -f
```

After regenerating the initrd, the crypto officer has to append the following string to the kernel command line by changing the setting in the boot loader:

```
fips=1
```

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"` respectively. For example:

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

The partition of /boot is located on /dev/sda1 in this example. Therefore, the following string needs to be appended to the kernel command line:

```
"boot=/dev/sda1"
```

Reboot to apply these settings.

FIPS 140-2 and AES-NI support

According to the kernel crypto API FIPS 140-2 security policy, the kernel crypto API module supports the AES-NI Intel processor instruction set as an approved cipher. The AES-NI instruction set is used by the kernel module.

In case you configured a full disk encryption using AES, you *may* use the AES-NI support for a higher performance compared to the software-only implementation.

To utilize the AES-NI support, the mentioned kernel module must be loaded during boot time by installing a plugin.

Before you install the plugin, you MUST verify that your processor offers the AES-NI instruction set by calling the following command:

```
        cat /proc/cpuinfo | grep aes
```

If the command returns a list of properties, including the I<aes> string, your CPU provides the AES-NI instruction set. If the command returns nothing, AES-NI is not supported.

You MUST NOT install the following plugin if your CPU does not support AES-NI because the kernel will panic during boot.

The support for the AES-NI instruction set during boot time is enabled by installing the following plugin (make sure that the version of the plugin RPM matches the version of the installed RPMs!):

```
        # install the dracut-fips-aesni package
        rpm -Uhv dracut-fips-aesni-*.noarch.rpm
        # recreate the initramfs image
        dracut -f
```

The changes come into effect during the next reboot.

Configuration Changes and FIPS Approved Mode

Use care whenever making configuration changes that could potentially prevent access to the fips_enabled flag (fips=1) in the file /proc/sys/crypto/fips_enabled. If the module does not detect this flag during initialization, it does not enable the FIPS approved mode.

All user space modules depend on this file for transitioning into FIPS 140-2 approved mode.

9.2 User Guidance

See the header file lib/libcryptsetup.h provided with the cryptsetup SRPM for developer information about the API.

See the cryptsetup(8) man page for general usage documentation for cryptsetup and LUKS extensions.

Except for the cryptographic officer accessing the volume key using cryptsetup, no other users shall access the volume key.

Besides cryptsetup, no other application may be used to manage dm-crypt module (either through libcryptsetup, libdevmapper,or using direct kernel dm-ioctl interface).

9.3 Handling Self Test Errors

OpenSSL, Kernel Crypto API, and libgcrypt self test failures may prevent dm-crypt from operating. See the Guidance section in those Security Policies for instructions on handling self test failures.

The dm-crypt self test consists of the software integrity test. If the integrity test fails, dm-crypt enters an error state. The only recovery from this type of failure is to reinstall the dm-crypt module including the kernel. If you downloaded the software, verify the package hash to confirm a proper download.

10 Mitigation of Other Attacks

10.1 NSS

The following is taken from the FIPS 140-2 Security Policy documents of the NSS module:

Other Attacks	Mitigation Mechanism	Specific Limitations
Cache-timing attacks on the modular exponentiation operation used in DSA	Cache invariant modular exponentiation This is a variant of a modular exponentiation implementation that Colin Percival [4] showed to defend against cache-timing attacks.	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.

11 Glossary and Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CBC	Cypher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CC	Common Criteria
CMVP	Cryptographic Module Validation Program
COTS	Commercial Off-the-Shelf
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESSIV	Encrypted Salt Sector for IV
FSM	Finite State Model
HMAC	Hash Message Authentication Code
LV	Logical Volume
LVM	Logical Volume Manager
NIST	National Institute of Science and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
O/S	Operating System
POST	Power-on Self Test
PP	Protection Profile
PV	Physical Volume
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SAP	Service Access Points
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SLA	Service Level Agreement
SOF	Strength of Function

SHS	Secure Hash Standard
SVT	Scenario Verification Testing
TDES	Triple-DES
TOE	Target of Evaluation
UI	User Interface

Table 8: Abbreviations

12 References

The following references are available on the CMVP website at http://csrc.nist.gov/groups/STM/cmvp/standards.html#02:

[1] National Institute of Standards and Technology, *FIPS PUB 140-2: Security Requirements for Cryptographic Modules*

[2] National Institute of Standards and Technology, *FIPS 140-2 Annex A: Approved Security Functions*

[3] National Institute of Standards and Technology, *FIPS 140-2 Annex B: Approved Protection Profiles*

[4] National Institute of Standards and Technology, *FIPS 140-2 Annex C: Approved Random Number Generators*

[5] National Institute of Standards and Technology, *FIPS 140-2 Annex D: Approved Key Establishment Techniques*

[6] National Institute of Standards and Technology and Communications Security Establishment, *Derived Test Requirements (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules*

[7] National Institute of Standards and Technology, *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication 46-3

[8] National Institute of Standards and Technology, *DES Modes of Operation*, Federal Information Processing Standards Publication 81

[9] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2

[10] National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, Federal Information Processing Standards Publication 180-1