Red Hat Enterprise Linux 7 OpenSSH Server Cryptographic Module

version rhel7.20190626

FIPS 140-2 Non-Proprietary Security Policy

Version 1.2

Last update: 2021-03-29

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

1 Introduction

This document is the non-proprietary Security Policy for the Red Hat Enterprise Linux 7 OpenSSH Server Cryptographic Module version rhel7.20190626. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 module.

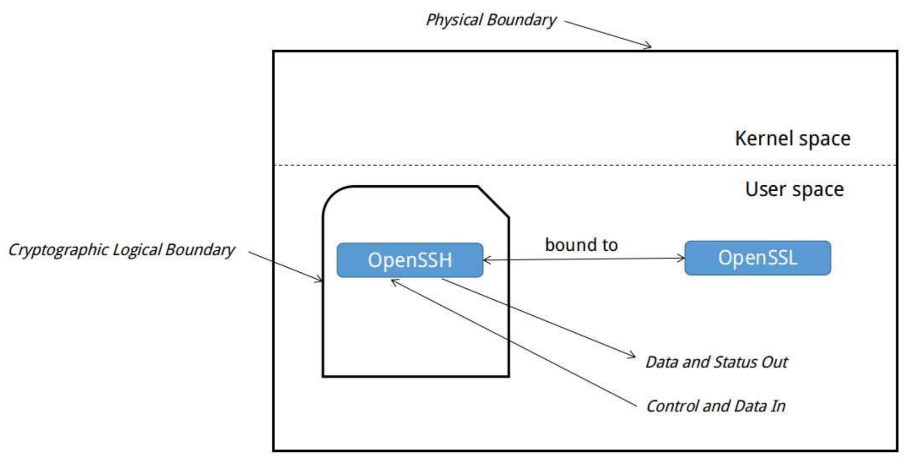
2 Cryptographic Module Specification

2.1 Module Overview

The Red Hat Enterprise Linux 7 OpenSSH Server Cryptographic Module version rhel7.20190626 (hereafter referred to as "the module") is a software library implementing the cryptographic support for the SSH protocol in the Red Hat Enterprise Linux user space.

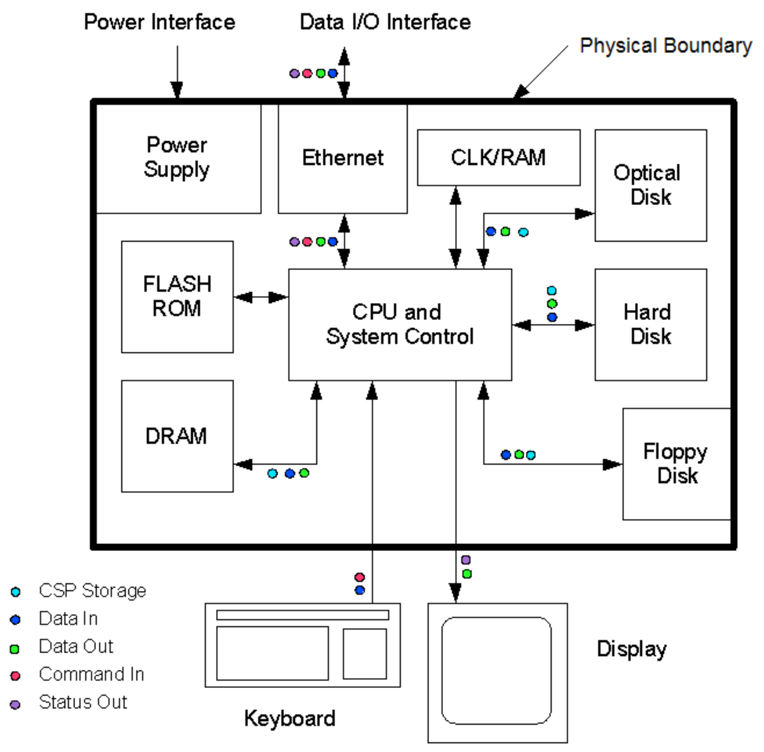
The module is implemented as a set of binary files.

Figure 1: Cryptographic Module Logical Boundary



The module is aimed to run on a general purpose computer; the physical boundary is the surface of the case of the target platform, as shown in the diagram below:

Figure 2: Cryptographic Module Physical Boundary



The module will use the Red Hat Enterprise Linux 7 OpenSSL Module (FIPS 140-2_Certificate #3867) as a bound module which provides the underlying cryptographic algorithms necessary for establishing and maintaining the SSH session. In addition the integrity check uses the cryptographic services provided by the Red Hat Enterprise Linux 7 OpenSSL Module as used by the utility application of fipscheck using the HMAC-SHA-256 algorithm.

This requires a copy of a Cert. #3867 validated version of the Red Hat Enterprise Linux 7 OpenSSL Module to be installed on the system for the current module to operate.

The cryptographic module combines a vertical stack of Linux components intended to limit the external interface each separate component may provide. The following software need to be installed for the module to operate:

- Red Hat Enterprise Linux 7 OpenSSH Server Cryptographic Module with the version of the OpenSSH server RPM file 7.4p1-21.el7
- The bound module of OpenSSL with FIPS 140-2 Certificate #3867
- The contents of the fipscheck RPM package (version 1.4.1-6.el7)
- The contents of the fipscheck-lib RPM package (version 1.4.1-6.el7).

The OpenSSH server RPM package of the Module includes the binary files, integrity check HMAC files and Man Pages. Any application other than the OpenSSH server application (/usr/sbin/sshd) delivered with the aforementioned OpenSSH RPM package is not part of the Module. The FIPS certificate for this module does not apply to these other applications.

The files comprising the module are the following:

- /usr/sbin/sshd
- /usr/bin/fipscheck
- /usr/lib64/fipscheck/sshd.hmac
- /usr/lib64/fipscheck/fipscheck.hmac
- /usr/lib64/fipscheck/libfipscheck.so.1.2.1.hmac
- /usr/lib64/libfipscheck.so.1.2.1.

2.2 FIPS 140-2 validation

For the purpose of the FIPS 140-2 validation, the module is a software-only, multi-chip standalone cryptographic module validated at Security Level 1. The table below shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard:

FIPS 140-2 Section		Security Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

Table 1: Security Levels

The module has been tested on the following platforms with the following configuration:

Constructor	Hardware	Processor	Operating System
Dell	PowerEdge R630	Intel(R) Xeon(R) CPU E5	Red Hat Enterprise Linux 7

Table 2: Tested Platforms

The physical boundary is the surface of the case of the target platform. The logical boundary is depicted in Figure 1: Cryptographic Module Logical Boundary.

The bound OpenSSL module includes algorithm implementations using the AES-NI Processor Algorithm Acceleration (PAA) functions provided by the processors.

2.3 Modes of Operations

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

The Module verifies the integrity of the runtime executable using a HMAC-SHA-256 digest operation and compares the value with the build time pre-computed value. If the digests match, the power-up self-tests are then performed. If the power-up self-tests are successful, the Module turns to FIPS Approved mode.

The following table shows algorithms available in FIPS mode and the services are listed can be found in section 4.2, Table 6.

The OpenSSH and the bound OpenSSL module together provide the Diffie Hellman and EC Diffie Hellman key agreement. The OpenSSH module only implements the KDF portion of the key agreement and the bound OpenSSL module provides the shared secret computation.

- Diffie-Hellman (CVL Certs. #C1378, #C1379, #C1385 and #C1386 with CVL Certs. #C1423, key agreement; key establishment methodology provides between 112 and 202 bits of encryption strength);
- EC Diffie-Hellman (CVL Certs. #C1378, #C1379, #C1385 and #C1386 with CVL Certs. #C1423, key agreement; key establishment methodology provides between 128 and 256 bits of encryption strength);

Algorithm	CAVS Certificates
Provided by OpenSSH Module	
SP 800-135 SSH KDF with SHA-1, SHA-256, SHA-384, SHA-512	CVL Cert. #C1423
Provided by bound OpenSSL Module	
AES (CBC, CTR, GCM) (128, 192, 256-bit keys) Note: OpenSSH only supports 128 and 256 bit keys for GCM	Certs. #C1378, #C1379, #C1380, #C1381, #C1382, #C1383, #C1384, #C1385, #C1386, #C1419
Triple-DES (CBC) (168-bit keys)	Certs. #C1378, #C1379, #C1385, #C1386
HMAC (SHA-1/224/256/384/512)	Certs. #C1378, #C1379, #C1380, #C1381, #C1382, #C1383, #C1384, #C1385, #C1386, #C1419
SHA (SHA-1/224/256/384/512)	Certs. #C1378, #C1379, #C1380, #C1381,#C1382, #C1383, #C1384, #C1385, #C1386, #C1419
RSA (2048, 3072-bit keys)	Certs. #C1378, #C1379, #C1385, #C1386
ECDSA	Cert. #C1378, #C1379, #C1385, #C1386

Algorithm	CAVS Certificates
(P-224/256/384/521)	
Diffie-Hellman shared secret computation with public key sizes between 2048 and 8192 bits	CVL Certs. #C1378, #C1379, #C1385, #C1386
EC Diffie-Hellman shared secret computation (P-256/384/521)	CVL Certs. #C1378, #C1379, #C1385, #C1386
DRBG (Hash, HMAC, and CTR)	Certs. #C1378, #C1379, #C1380, #C1381, #C1382, #C1383, #C1384, #C1385, #C1386, #C1419
NDRNG	Non-approved but allowed used for seeding DRBG

Table 3: Approved or Allowed Algorithms

The following table lists the non-approved algorithms, use of any of these algorithm will put the module in non FIPS mode.

Algorithm	Notes
RSA	Using keys less than 2048 bits
DSA	OpenSSH only supports 1024-bit keys

Table 4: Non Approved Algorithms from bound OpenSSL module

3 Cryptographic 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 following table summarizes the four logical interfaces:

Logical interfaces	Description	Physical ports mapping the logical interfaces
Command In	Invocation of the sshd command on the command line or via the configuration file /etc/ssh/sshd_config Environment variable SSH_USE_STRONG_RNG	Keyboard, Ethernet port
Status Out	Status messages returned after the command execution	Display, Ethernet port
Data In	Input parameters of the sshd command on the command line with host key files in /etc/ssh, ~/.ssh/authorized_keys, locally stored data, data via SSHv2 channel, data via local or remote port-forwarding port	Keyboard, Ethernet port
Data Out	Output data returned by the sshd command	Display, Ethernet port

Table 5: Ports and Logical Interfaces

4 Roles, Services and Authentication

4.1 Roles

The module supports the following roles:

- **User role**: performs Key Derivation Function, Establish & Maintain SSH Session, Close SSH Session and Show Status
- **Crypto Officer role**: performs module installation and configuration, perform the self-tests and terminate SSH Application

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

4.2 Services

The module supports services available to users in the available roles. All services are described in detail in the user documentation.

The following table shows the available services, the roles allowed, the Critical Security Parameters (CSPs) involved and how they are accessed in the FIPS mode.

'R' stands for Read permission, 'W' stands for write permission and 'EX' stands for executable permission of the module:

Service	Algo(s).	Note(s)	Role	CSPs	Access
Establish & Maintain SSH Session	SP 800-135 Key Derivation Function in the SSH protocol version 2	N/A	User	RSA or ECDSA server private key Diffie-Hellman or EC Diffie-Hellman shared secret, derived keys Derived session encryption keys and derived data authentication (HMAC) keys	R, W, EX
Close SSH Session	N/A	Zeroize	User	Derived session encryption key and derived data authentication (HMAC) keys Shared secret	W
Terminate SSH Application	N/A	Zeroize	Crypto officer	Derived session encryption key and data authentication (HMAC) keys, Shared secret	W
Self-Tests	HMAC-SHA-256 (uses the cryptographic services provided by the bound OpenSSL module)	Integrity test invoked by restarting the module	Crypto officer	HMAC integrity key	R, EX
Zeroize	N/A	N/A	User	All aforementioned CSPs	W
Show Status	N/A	Via verbose mode and exit codes	User	N/A	N/A
Configure SSH Server	N/A	N/A	Crypto officer	N/A	N/A
Installation	N/A	N/A	Crypto officer	N/A	N/A

Table 6: Available Cryptographic Module's Services in FIPS mode

Note: The SSH protocol has not been reviewed or tested by the CAVP and CMVP. Only the SP 800-135 Key Derivation Function has been validated by CAVP.

Service	Algo(s).	Note(s)	Role	CSPs	Access
Establish & Maintain SSH Session	SP 800-135 Key Derivation Function in the SSH protocol version 2	N/A	User	Using RSA or DSA keys listed Table 4.	R, W, EX

Table 7: Available Cryptographic Module's services in Non-FIPS mode

4.3 Authentication

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

5 Physical Security

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

6 Operational Environment

6.1 Applicability

The module operates in a modifiable operational environment per FIPS 140-2 Security Level 1 specifications. The module runs on a commercially available general-purpose operating system executing on the hardware specified in section 2.2.

The Red Hat Enterprise Linux operating system is used as the basis of other products which include but are not limited to:

- Red Hat Enterprise Linux Atomic Host
- Red Hat Virtualization (RHV)
- Red Hat OpenStack Platform
- OpenShift Container Platform
- Red Hat Gluster Storage
- Red Hat Ceph Storage
- Red Hat CloudForms
- Red Hat Satellite.

Compliance is maintained for these products whenever the binary is found unchanged.

6.2 Policy

The operating system is restricted to a single operator (concurrent operators are explicitly excluded). The entity using the application is the single user of the module, even when the application is serving multiple clients.

In the operational mode, the ptrace(2) system call, the debugger (gdb(1)), and strace(1) shall be not used.

7 Cryptographic Key Management

7.1 Random Number Generation

The module does not implement any random number generator nor provides key generation. The module only provides key derivation through the implementation of the SP 800-135 KDF.

The module calls the bound OpenSSL module to obtain the shared secret which will be used during the SSHv2 protocol initial handshake. The module derives keys from this shared secret through the SP 800-135 KDF implementation. When the module requests encryption/decryption services provided by the OpenSSL bound module, the resulting derived symmetric key (i.e. the output of the SP 800-135 KDF) will be passed to the OpenSSL bound module via API parameters.

Here are listed the CSPs/keys details concerning storage, input, output, generation and zeroization:

Type	Keys/CSPs	Key Generation	Key Storage	Key Entry/Output	Key Zeroization
Session Encryption Keys	Shared secret (2048 bits to 8192 bits for Diffie-Hellman; P-256, P-384, P-521 for EC Diffie-Hellman)	N/A	Module's memory	Entry via API parameters Output: N/A	Zeroized by the sshd application
	Derived keys (AES 128/192/256-bit keys; Triple-DES 168-bit keys; HMAC keys larger than 112 bits)	(Derived from the shared secret through the SP 800-135 KDF)	Module's memory	Entry: N/A Output via API parameters	
Server Private Keys	RSA private keys (2048/3072/4096-bit keys)	N/A	Module's memory	Via API parameters	Zeroized by the sshd application
	ECDSA private keys (P-256, P-384, P-521 keys)	N/A	Module's memory	Via API parameters	Zeroized by the sshd application
Software Integrity Key	HMAC Key (128-bit key)	N/A	Module's binary file	N/A	N/A

Table 8: Keys/CSPs

7.2 Key / CSP Storage

Public and private keys are provided to the module by the calling process, and are destroyed from memory when released. The module does not perform persistent storage of keys. The keys and CSPs are temporarily stored as plaintext in the RAM.

The persistently stored public keys that are associated with a client username via the use of the files ~/.ssh/id_rsa and ~/.ssh/id_ecdsa in which they are both stored for each user individually in its home directory. These files however, are not part of the module.

The persistent storage of private host keys is in /etc/ssh/ssh_host_rsa_key and /etc/ssh/ssh_host_ecdsa_key. These files however, are not part of the module.

7.3 Key / CSP Zeroization

The destruction functions overwrite the memory occupied by keys with pre-defined values and deallocates the memory with the free() call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten before the physical memory is allocated to another process.

8 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

MARKETING NAME..........................PowerEdge R630
REGULATORY MODEL....................E26S
REGULATORY TYPE........................E26S001
EFFECTIVE DATE............................September 03, 2014
EMC EMISSIONS CLASS.................Class A

8.1 Statement of compliance

This product has been determined to be compliant with the applicable standards, regulations, and directives for the countries where the product is marketed. The product is affixed with regulatory marking and text as necessary for the country/agency. Generally, Information Technology Equipment (ITE) product compliance is based on IEC and CISPR standards and their national equivalent such as Product Safety, IEC 60950-1 and European Norm EN 60950-1 or EMC, CISPR 22/CISPR 24 and EN 55022/55024. Dell products have been verified to comply with the EU RoHS Directive 2011/65/EU. Dell products do not contain any of the restricted substances in concentrations and applications not permitted by the RoHS Directive.

9 Self-Tests

9.1 Power-Up Self-Tests

The module performs power-up self-tests at module initialization to ensure that the module is not corrupted. The self-tests are automatically triggered without any user intervention.

While the module is performing the power-up tests, services are not available, and input or output data is not possible: the module is single-threaded and will not return to the calling application until the self-tests are completed successfully.

9.1.1 Integrity Tests

The integrity check is performed by the fipscheck application using the HMAC-SHA-256 algorithm implemented by the bound Red Hat Enterprise Linux 7 OpenSSL Module.

When the OpenSSH module starts, it triggers the power-on self-tests, including the software integrity test. The software integrity test, using the HMAC-SHA-256 algorithm, constitutes a known answer test for the HMAC-SHA-256 algorithm.

The user space integrity verification is performed as follows: the OpenSSH Server application links with the library libfipscheck.so which is intended to execute fipscheck to verify the integrity of the OpenSSH server application file using the HMAC-SHA-256 algorithm. Upon calling the FIPSCHECK_verify() function provided with libfipscheck.so, fipscheck is loaded and executed, and the following steps are performed:

1. OpenSSL, loaded by fipscheck, performs the integrity check of the OpenSSL library files using the HMAC-SHA-256 algorithm
2. fipscheck performs the integrity check of its application file using the HMAC-SHA-256 algorithm provided by the OpenSSL Module
3. fipscheck automatically verifies the integrity of libfipscheck.so before processing requests of calling applications
4. The fipscheck application performs the integrity check of the OpenSSH server application file. The fipscheck computes the HMAC-SHA-256 checksum of that and compares the computed value with the value stored inside the /usr/lib64/fipscheck/sshd.hmac checksum file. The fipscheck application returns the appropriate exit value based on the comparison result: zero if the checksum is OK, an error code otherwise (which brings the OpenSSH Module into the error state). The libfipscheck.so library reports the result to the OpenSSH server application.

If any of those steps fail, an error code is returned and the OpenSSH Module enters the error state.

9.1.2 Cryptographic algorithm tests

The power-up self tests for the SP 800-135 KDF is covered by the SHS Known-Answer-Tests performed by the bound Red Hat Enterprise Linux 7 OpenSSL Module.

9.1.3 On-demand self-tests

The user can perform on-demand self-tests by restarting the sshd service.

10 Guidance

The following guidance items are to be used for assistance in maintaining the module's validated status while in use.

10.1 Crypto Officer Guidance

The version of the RPMs containing the FIPS validated Module is stated in section 2.1 above. The Red Hat Enterprise Linux 7 OpenSSL Module referenced in section 2.1 must be installed according to its Security Policy.

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.

1 Disable the prelink:

```
# sed -i 's/PRELINKING=yes/PRELINKING=no/g' /etc/sysconfig/prelink
```

2 Run following command to return binaries to a non-prelink state:

```
# /usr/sbin/prelink -ua
```

Only the cipher types listed in section 1.2 are allowed to be used.

Crypto officer should perform the following steps for Module initialization :

1. Install the dracut-fips package:

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

2. Recreate the INITRAMFS image:

```
# dracut -f
```

After regenerating the initramfs, 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.

The next step is to check the presence of the configuration file /proc/sys/crypto/fips_enabled and make sure it contains value 1.

2.1.1 The version of the RPM containing the validated Module is the version listed in chapter 2.1. 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 RPM tool indicates an integrity error.

10.1.1 OpenSSH Configuration

The user must not use DSA keys for performing key-based authentication as OpenSSH only allows DSA keys with 1024 bit size which are disallowed as per SP800-131A.

The user must not accept DSA host keys potentially offered during the first contact of an SSH server as OpenSSH only allows DSA keys with 1024 bit size which are disallowed as per SP800- 131A.

When re-generating RSA host keys, the crypto officer should generate RSA keys with a size of 2048 bit or higher according to [SP800-131A]. The crypto officer should inform the user base to not use RSA keys with key sizes smaller than 2048 bits.

In FIPS 140-2 mode, the following restrictions are applicable. When these restrictions are violated by configuration options or command line options, the module will not be in the FIPS mode of operation:

- SSH protocol version 1 is not allowed
- GSSAPI is not allowed
- Only the following ciphers are allowed:
  - aes128-ctr
  - aes192-ctr
  - aes256-ctr
  - aes128-cbc
  - aes192-cbc
  - aes256-cbc
  - aes128-gcm@openssh.com
  - aes256-gcm@openssh.com
  - 3des-cbc
  - rijndael-cbc@lysator.liu.se

Only the following message authentication codes are allowed:

- hmac-sha1
- hmac-sha2-256
- hmac-sha2-512
- hmac-sha1-etm@openssh.com
- hmac-sha2-256-etm@openssh.com
- hmac-sha2-512-etm@openssh.com

Any use of other ciphers or algorithms will results in the module entering the non-FIPS mode of operation.

10.2 User Guidance

Use the 'systemctl start sshd' command to start the OpenSSH server, or configure the server to start using 'Systemctl enable/disable'.

This module is used by connecting to it with a ssh client. See the documentation of the client, e.g. the Red Hat Enterprise Linux 7.7 OpenSSH Client Cryptographic Module's Security Policy and the sshd(1) man page, for more information.

10.2.1 Handling Self-Test Errors

OpenSSL's self tests failures may prevent OpenSSH from operating. See the Guidance section in the OpenSSL Security Policy for instructions on handling OpenSSL self test failures.

The OpenSSH self test consists of the software integrity test. If the integrity test fails, OpenSSH enters an error state. The only recovery from this type of failure is to reinstall the OpenSSH module.

10.2.2 AES-GCM

IV generation for AES-GCM only occurs in the context of the SSHv2 protocol. The module is compliant with RFC4252, 4253 and 5647.

When an SSH session gets terminated for any reason, all keying material will be re-negotiated by the module.

The module enforces a maximum of 2^{31} packets that can be either sent or received by the module for a given SSH session, which satisfied the 2^{64}-1 AES-GCM encryption limit imposed by IG A.5.

Appendix A Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
CVT	Component Verification Testing
DES	Data Encryption Standard
DFT	Derivation Function Test
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECC	Elliptic Curve Cryptography
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
FSM	Finite State Model
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NDRNG	Non-Deterministic Random Number Generator
OFB	Output Feedback
O/S	Operating System
PAA	Processor Algorithm Acceleration
PR	Prediction Resistance
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
TDES	Triple DES
UI	User Interface

Appendix B References

FIPS180-4	**Secure Hash Standard (SHS)** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS197	**Advanced Encryption Standard** November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/detail/fips/198/1/final
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-56A	**NIST Special Publication 800-56A Revision 2 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** May 2013 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800 56Ar2.pdf
SP800-56C	**Recommendation for Key Derivation through Extraction-then-Expansion** November 2011 http://csrc.nist.gov/publications/nistpubs/800-56C/SP-800-56C.pdf
SP800-90A Rev. 1	**NIST Special Publication 800-90A Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final
SP800-90B	**NIST Draft Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://csrc.nist.gov/publications/detail/sp/800-90b/final
SP800-108	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions** October 2009 http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf
SP800-131A	**NIST Special Publication 800-131A Revision 2 - Transitions:**

Rev. 2	**Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final