Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module v1.0

FIPS 140-2 Security Policy

version 1.7

Last Update: 2010-07-07

Contents

Document History

Version	Date of Change	Author	Changes to Previous Version
0.1	2008-07-03	Steve Weingart atsec	Initial version
1.0	2009-11-04	Steve Weingart atsec	First release version
1.1-pre	2009-11-11	Steve Weingart atsec	Correction of modutil call
1.2	2009-12-21	Steve Weingart atsec	Add policy to guidance
1.3	2010-02-12	Steve Weingart atsec	Update reference to OpenSWAN package RPM which includes bugfix RHBA:2010-0096
1.4	2010-04-13	Steve Weingart atsec	Update single user mode statement
1.5	2010-06-14	Steve Weingart atsec	Update of NSS CAVS certificates
1.6	2010-06-30	Steve Weingart atsec	Update of SW block diagram
1.7	2010-07-07	Steve Weingart atsec	Clarification of the CAVS certificate listing

1 Cryptographic Module Specification

This document is the non-proprietary security policy for the Red Hat Enterprise Linux 5 OpenSwan 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.

Description of Module

The Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module is a software only cryptographic module that provides the IKE protocol version 1 and version 2 key agreement services required for IPSec. The OpenSwan module is a software only, security level 1 cryptographic module, running on a multi-chip standalone platform.

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	N/A

Table 1: Security Levels

The module has been tested on the following platforms:

Manufacturer	Model	O/S & Ver.
HP	HP Integrity Server RX2660	Red Hat Enterprise Linux 5.4 (Single User Mode)
HP	HP ProLiant Server DL585	Red Hat Enterprise Linux 5.4 (Single User Mode)

Table 2: Platforms Tested

This cryptographic module combines a vertical stack of Linux components intended to limit the external interface each separate component may provide. The list of components and the cryptographic boundary of the composite module is defined as follows:

- Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module with the version of the RPM file of openswan-2.6.21-5.el5_4.3.
- Network Security Service (NSS), a separately validated cryptographic module (FIPS 140-2 certificate

#815) provides cryptographic algorithms and security functions for the Pluto IKE Daemon. The OpenSwan module uses this module in accordance with the Security Rules stated in the *NSS Cryptographic Module Version 3.11.4 Security Policy.* The RPM file that contains all of the files for the validated version of the Red Hat Enterprise Linux NSS Cryptographic module is version nss-3.12.6-2.el5_4. The vendor affirmation covering the tested hardware platforms can be found at: http://www.redhat.com/solutions/government/certifications/. Note: The NSS version subject to validation was 3.11.4. As the NSS FIPS140-2 certificate does not cover the IA64 hardware architecture, the source code was recompiled, without any change, for the IA64 hardware platform. This is consistent with the vendor affirmation requirements in the FIPS 140-2 Implementation Guidance, G.5 item 1) a) i).

- The module integrity check is performed by the Red Hat Enterprise Linux utility fipscheck supported by the associated library which is linked with the Pluto IKE Daemon. The version of the utility and its library is 1.2.0-1.el5.
- The cryptographic services for the fipscheck utility are provided by the OpenSSL Cryptographic Module for standard cryptographic operations in the FIPS 140-2 level 1 validated version (FIPS 140-2 validation certificate #1320). This OpenSSL library provides the HMAC SHA-256 cryptographic mechanisms. The RPM file that contains all of the files for the validated version of the Red Hat Enterprise Linux OpenSSL Cryptographic module, used for the module integrity test, is version 0.9.8e-12.el5.

This cryptographic boundary limits the external interface of the module to the interface provided by the Pluto daemon for the purposes of IKE. Therefore, it eliminates the need to protect cryptographic keys and other CSPs when crossing inter-component boundaries internal to the module.

The module is a FIPS security level 1 software module. The Linux platform is constrained to be used by a single user.

Description of Approved Mode

The Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module cryptographic boundary is defined by the following ciphers:

Approved Algorithms provided by the NSS library (CMVP Certificate #815 with all other available cipher mechanisms are not enabled in the FIPS mode of the Pluto IKE Daemon):

- Triple-DES (Cert# 943)
- AES (Cert# 1368)
- SHA-1 (Cert# 1250)
- RNG (Cert# 755)
- DSA (Cert# 449)
- RSA (Cert# 669)

Approved Algorithms provided by the OpenSSL library:

- HMAC SHA-256 (Certs #661, #662 and #663)

Caveat: The module will support the following non-approved functions.

- Diffie-Hellman (key agreement, key establishment methodology provides between 80 bits and 112 bits of encryption strength)
- EC Diffie-Hellman (key agreement, key establishment methodology provides between 80 bits and 256 bits of encryption strength)

- RSA (key wrapping, key establishment methodology provides between 80 bits and 192 bits of encryption strength)

The NSS library implements the following non-Approved algorithms, which shall not be used in the FIPS Approved mode of operation:

- RC2 , RC4, or DES for symmetric key encryption and decryption.
- MD2 or MD5 for hashing.

Cryptographic Module Boundary

The Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module physical boundary is defined by the surface of the case of the platform tested on. The logical module boundary is depicted in the software block diagram and is embodied by the Pluto IKE Daemon and its supportive applications found at /usr/libexec/ipsec/ which link with the NSS library. The integrity check mechanism provided by the fipscheck application and the OpenSSL library are part of the cryptographic module but not depicted as they are only used during load time of the module.

1.1.1 Hardware Block Diagram

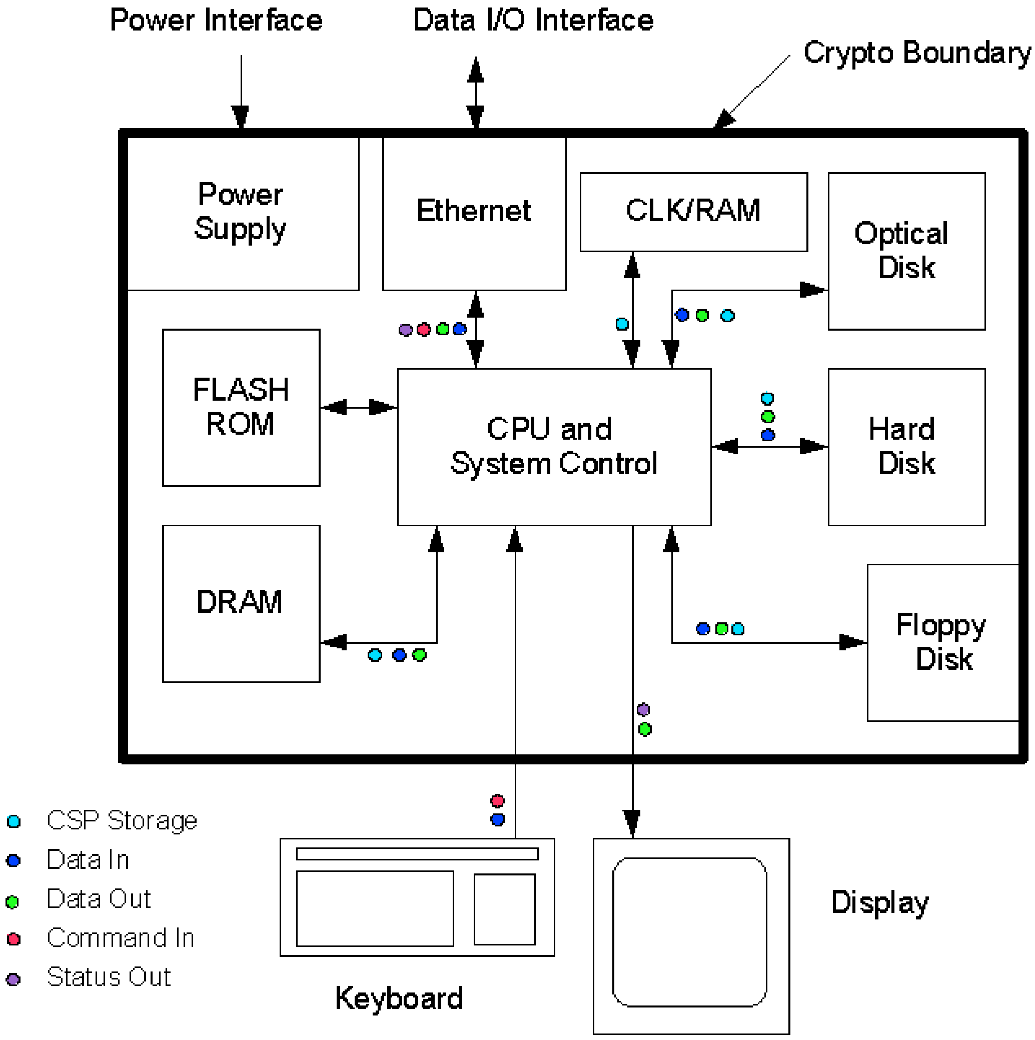

Figure 1: Hardware Block Diagram

1.1.2 Software Block Diagram

Note: This security policy only covers the user space module which includes the parts above the User/Kernel line in the block diagram below.

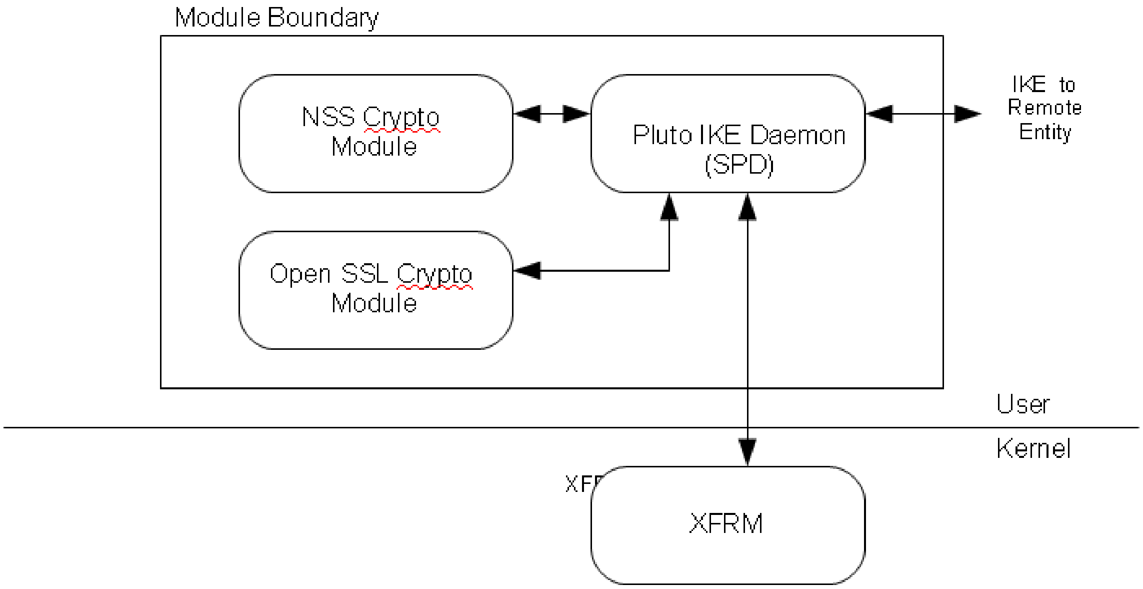

Figure 2: Software Block Diagram

2 Cryptographic Module Ports and Interfaces

Function	Port
Control In	Network Port/Protocol, Configuration Files (/etc/ipsec.conf, /etc/ipsec.secrets), Linux Kernel (XFRM Interface), command line
Status Out	Log File, Network Port/Protocol
Data In	Network Port/Protocol, NSS Key Database file stored in /etc/ipsec.d/
Data Out	Network Port/Protocol, Linux Kernel (XFRM Interface)

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.

Roles

Role	Services (see list below)
User	Encryption, Decryption (symmetric and public/private), Random Numbers
Crypto Officer	Configuration, Encryption, Decryption (symmetric and public/private), Random Numbers

Table 4: Roles

The user role is assumed by the underlying server application that makes calls to the module on behalf of one or more external clients [Reference: Implementation Guidance for FIPS PUB 140-2, dated 5-22-08, Section 6.1].

Services

The module supports services that are available to users in the various roles. All of the services are described in detail in the module's user documentation. The following table shows the services available to the various roles.

Service	**Cryptographic Keys and CSPs Accessed**	**Crypto Officer**	**User**
Install and Configure the module.	RSA public/private keys are added for SPD	●	
Manage Pluto IKE Daemon start, stop, etc.	DRNG Seed and Seed Key Zeroize of CSPs, Keys	●	
Negotiate IKE to establish security associations	DH private and public parameters RSA public/private used for authentication ISAKMP SA encryption key IPSEC SA encryption key	●	●
Run the FIPS self test (initiation by restarting the module)	N/A	●	
Read FIPS Status	N/A	●	

Table 5: Operational Services

Operator Authentication

There is no operator authentication, assumption of role is implicit by action.

Mechanism and Strength of Authentication

No authentication is required at security level 1, authentication is implicit by assumption of the role.

4 Physical Security

This is a security level 1, software only module and does not claim any physical security.

5 Operational Environment

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

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.

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

6 Cryptographic Key Management

This section describes how keys and critical security parameters (CSP) are handled by 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 by the module via NSS.

Key life cycle table

Key	Type	Generation	Establishment	Access by Service	Entry and output method	Storage	Zeroization
RSA Private and Public Keys	RSA key	N/A	N/A	Authentication in the ISAKMP SA negotiation	N/A	Plaintext	Immediately after use by NSS
ISAKMP Security Association Tunnel Encryption Keys	AES or Triple-DES	N/A	Established during the ISAKMP SA handshake using DH.	Establish & Maintain ISAKMP SA	N/A	Ephemeral	Close of ISAKMP SA or termination of Pluto IKE Daemon
IPSEC Security Association Tunnel Encryption Keys	AES or Triple-DES	N/A	Established during the IPSEC SA handshake using DH.	Establish & Maintain IPSEC SA	Transfer to the Linux Kernel via XFRM Interface	Ephemeral	Close of ISAKMP SA or overwritten by re-negotiated IPSEC SA or termination of Pluto IKE Daemon

Key	Type	Generation	Establishment	Access by Service	Entry and output method	Storage	Zeroization
Diffie-Hellman Private and Public Parameters	DH	ANSI FIPS 186-2 RNG	N/A	Establish & Maintain ISAKMP SA and IPSEC SA	N/A	Ephemeral	Close of ISAKMP SA or termination of Pluto IKE Daemon
RNG Seed	random value	N/A	N/A	Establish & Maintain ISAKMP SA and IPSEC SA	N/A, provided by /dev/urandom	Ephemeral	N/A (Termination of Pluto IKE Daemon where NSS zeroizes seed)
RNG Seed Key	random value	N/A	N/A	Establish & Maintain ISAKMP SA and IPSEC SA	N/A, provided by /dev/urandom	Ephemeral	N/A (Termination of Pluto IKE Daemon where NSS zeroizes key)
Software Integrity Key for OpenSSL, fipscheck and all Pluto IKE applications	HMAC SHA-256	N/A	N/A	Self-Tests	N/A	Plaintext within the OpenSSL and fipscheck libraries	Termination of the fipscheck application
Software Integrity Key for NSS library	DSA	N/A	N/A	Self-Tests	N/A	Plaintext within the NSS library	Termination of Pluto IKE Daemon

Table 6: Key Life Cycle

Notes:

Private keys are always encrypted by the NSS library. When an operation, requires a private key, the first pointer or handle to the private key is obtained using the public key and CKAID. Only during the operation private keys are decrypted and the operation is performed. After the operation is over, the memory pointing to the private key is zeroized by NSS. Until the private keys from the NSS database need to be deleted, there is special zeroization required – for details about the maintenance of keys by the NSS library, see CMVP certificate #815.

Key Zeroization

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

Random Number Generation

The module employs a FIPS186-2 x-Change Notice and a FIPS186-2 General Purpose x-Change Notice PRNG provided by the NSS library, which is seeded by the kernel.

The Linux kernel provides /dev/urandom as a source of random numbers for RNG seeds. The Linux kernel initializes this pseudo device at system startup.

The kernel performs continual tests on the random numbers it uses to ensure that the seed and seed key input to the Approved RNG do not have the same value. The kernel also performs continual tests on the output of the approved RNG to ensure that consecutive random numbers do not repeat.

7 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

Product Name and Model: HP ProLiant Server DL585 Series
Regulatory Model Number: HSTNS-1025
Product Options: All
conforms to the following Product Specifications and Regulations:
EMC: Class A
CISPR 22:2005
EN 55022:2006
EN 55024:1998 +A1:2001 +A2:2003
EN 61000-3-2:2006
EN 61000-3-3:1995 +A1:2001 +A2:2005

Product Name and Model: HP Integrity Server rx2660
Regulatory Model Number: RSVLA-0503
Product Options: All
conforms to the following Product Specifications and Regulations :
EMC: Class A
CISPR22:1997 / EN 55022:1998
CISPR 24:1997 + A1:2001 + A2: 2002 / EN 55024:1998 + A1:2001 + A2:2003
EN 61000-3-2:2000
EN 61000-3-3:1995 +A1:2001

8 Self Tests

FIPS 140-2 requires that the module perform self tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition some functions require continuous verification of function, such as the random number generator. All of these tests are listed and described in this section.

The module performs both power-on self test (POST) and conditional self tests to verify the integrity and correct operational functioning of the cryptographic module. If the system fails a self test, it reports status indicating that a failure has occurred and transitions to an error state, blocking all data input, data output and control input via their respective interfaces.

While the module is performing any power on self test or conditional test, software rules within the executable image prevent the module from entering a state where data output via the data output interface is possible.

The crypto officer with physical or logical access to the module can run the POST on demand by power cycling the module or by rebooting the operating system.

The following table summarizes the system self tests and conditional tests.

Self Test	Description
Mandatory power-up tests performed at power-up and on demand:	
Cryptographic Algorithm Known Answer Tests	Each cryptographic algorithm (see section1.1 for algorithm list) performed by the module, is tested using a “known answer” test to verify the correct operation of the algorithm. These tests are performed by the NSS library before it makes itself available to the Pluto IKE Daemon. In addition, the HMAC SHA-256 KAT is performed by the Red Hat Enterprise Linux OpenSSL library. HMAC SHA-256 is only used for the integrity check.
Integrity Test	The module computes an HMAC SHA-256 value for the the OpenSSL library, the fipscheck utility and all applications forming the OpenSwan (this) module and compares it to a pre-calculated value stored within the system. The integrity check is performed by the Red Hat Enterprise Linux OpenSSL Cryptographic Module utility fipscheck using OpenSSL for the HMAC SHA-256 implementation. The integrity verification of the NSS library is performed by the NSS library using DSA.
Critical Functions tests performed at power-up:	
None	No security-relevant critical functions tests are performed.
Conditional tests performed, as needed, during operation:	
Continuous RNG	16 bits continuous testing is performed during each use of the approved RNG. This test is a “stuck at” test to check the RNG output data for failure to a constant value.

Table 7: Self Tests

Any self test success or failure messages are output to the Pluto IKE Daemon error log file.

Known answer tests for encryption/decryption or hashing, function by encrypting or hashing a string for which the calculated output is known and stored within the cryptographic module. An encryption or hashing test passes when the freshly calculated output matches the expected (stored) value. A test fails when the calculated output does not match the expected value. For decryption, the test then decrypts the ciphertext encrypted string. A decryption test passes when the freshly calculated output matches the plaintext value. A decryption test fails when the calculated output does not match the plaintext value.

Known answer tests for Random Number Generators function by seeding the RNG with known values and checking that the output matches the pre-calculated value stored within the cryptographic module. The test passes when the freshly generated output matches the pre-calculated value. A test fails when the generated output does not match the pre-calculated value.

9 Guidance

The following section provides guidance for the Crypto Office for using the module in a way that maintains compliance with FIPS 140-2.

No specific guidance for the user is to be provided as the user actions do not have an impact to the maintenance of a secure operational state of the module.

Cryptographic Officer Guidance

NOTE: All cryptographic functions for the Red Hat Enterprise Linux 5 OpenSwan Cryptographic Module will be provided by a copy of a FIPS 140-2 validated version of the Red Hat NSS library. The OpenSSL library is used to perform integrity verification.

- Configure pluto as specified in ipsec.conf(5) and ipsec.secrets(5) man pages as well as the file README.nss provided by the OpenSwan RPM package.
- The correct policy for the RHEL5.4 validated version is selinux-policy-2.4.6-255.el5_4.2.noarch
- To start and stop the module, use the (service ipsec) command.
- Stopping the module will zeroize the ephemeral CSPs and keys.
- To check FIPS 140-2 module status, read the pluto debug data using the ipsec_barf(8) tool.
- The version of the RPM containing the validated module is stated in section1.1 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.
- Pre-shared Keys are not supported and shall not be used in FIPS mode.
- Only the FIPS 140-2 approved and allowed ciphers listed in section 1.1 shall be used in configuring the pluto daemon.
- When zeroizing the module the crypto officer is responsible for using a FIPS140-2 approved mechanism to clear the keys written on disk.
- The database for the cryptographic keys used by the pluto daemon must be initialized after it has been created as documented in the README.nss documentation with the following command assuming that the database is stored in the directory /etc/ipsec.d/
  - `modutil -fips true -dbdir /etc/ipsec.d`

NOTE: Encryption and decryption of data is done implicitly when the kernel triggers pluto to set up a new Security Association.

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.

To bring the module into FIPS mode, the crypto officer has to regenerate the initrd by using the following command:

For the x86_64 platform, the command is:

```
mkinitrd --with-fips -f /boot/initrd-$(uname -r).img $(uname -r)
```

For the IA64, the command is:

```
mkinitrd --with-fips -f /boot/efi/efi/redhat/initrd-$(uname -r).img $(uname -r)
```

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
```

This operation causes the FIPS flag to be set by the kernel which can be read in /proc/sys/crypto/fips_enabled (if the file contains a 1, the FIPS mode is enabled – a 0 specifies a disabled FIPS mode).

10 Mitigation of Other Attacks

No other attacks are mitigated.

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
CFB	Cypher Feedback
CC	Common Criteria
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECB	Electronic Code Book
FSM	Finite State Model
HMAC	Hash Message Authentication Code
LDAP	Lightweight Directory Application Protocol
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
O/S	Operating System
PP	Protection Profile
RNG	Randome Number Generator
RSA	Rivest, Shamir, Addleman
SAP	Service Access Points
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SOF	Strength of Function
SSH	Secure Shell

SVT	Scenario Verification Testing
TDES	Triple DES
TOE	Target of Evaluation
UI	User Interface

Table 8: Abbreviations

12 References

[1] Open Swan user guide (provided with installation RPM, see section 1.1 Description of Module for version)

[2] rx2660_EMIEMC_cert.pdf (On file at Red Hat)

[3] DL585_EMIEMC_CEcert.pdf (On file at Red Hat)

[4] FIPS 140-2 Standard, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[5] FIPS 140-2 Implementation Guidance, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[6] FIPS 140-2 Derived Test Requirements,http://csrc.nist.gov/groups/STM/cmvp/standards.html

[7] FIPS 197 Advanced Encryption Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[8] FIPS 180-3 Secure Hash Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[9] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/PubsFIPS.html

[10] FIPS 186-3 Digital Signature Standard (DSS), http://csrc.nist.gov/publications/PubsFIPS.html

[11] ANSI X9.52:1998 Triple Data Encryption Algorithm Modes of Operation, http://webstore.ansi.org/FindStandards.aspx?Action=displaydept&DeptID=80&Acro=X9&DpName=X9,%20Inc.