Symantec Corporation

Symantec Cryptographic Module

SW Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.0

Prepared for:

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

Phone: (408) 517-8000
http://www.symantec.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030

Phone: (703) 267-6050
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Symantec Cryptographic Module from Symantec Corporation. This Security Policy describes how the Symantec Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The Symantec Cryptographic Module is referred to in this document as SymCrypt, the cryptographic module, or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Symantec website (http://www.symantec.com) contains information on the full line of products from Symantec.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence Document
- Finite State Model
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Symantec. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Symantec and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Symantec.

2 SymCrypt

This section describes the Symantec Cryptographic Module from Symantec Corporation.

2.1 Overview

As one of the world's largest software companies, Symantec offers a comprehensive portfolio of security, storage, and systems management solutions. Symantec has a customer base that includes 99% of Fortune 1000 companies. It also has a broad range of product offerings that range from consumer virus protection to enterprise class Security Operation Centers (SOCs) that are staffed 24/7. Through their software and services, they help consumers and organizations protect information and infrastructure against more risks at more points, more completely and efficiently than any other company. With an arsenal that includes engineering centers, global patents, and cutting-edge research, customers can have what only a global leader like Symantec can provide - confidence in a connected world.

2.1.1 Symantec Security Information Manager

The first Symantec solution to take advantage of SymCrypt is the Symantec Security Information Manager. SSIM is a high availability enterprise class software solution, whose primary purpose is to preempt or detect security incidents while providing the framework to both respond and demonstrate compliance. SSIM accomplishes this through its integrated log management, distributed architecture, and automated updates from Symantec's Global Information Network (GIN), which offers real-time intelligence on the latest vulnerabilities and threats from around the world.

SSIM collects security information, called events, from a broad range of applications, services, and security products. It then converts that information into actionable intelligence by using its built-in asset management function for prioritization, and then applying its sophisticated rule-based correlation engine on a normalized event stream. Event data is easily managed and quickly retrieved using SSIM's specialized form of event detail storage, which uses proprietary indexing and compression. Large amounts of diverse event data can be centralized in online or archived event stores using direct-attached storage (DAS), network-attached storage (NAS) or storage area network (SAN). SSIM also embeds a high performance relational database to store both summarized events and data pertaining to incidents, tickets, assets, rules, vulnerabilities, workflow, and reports. This allows for trend reporting and custom SQL[1] queries, with drill down capability into the appropriate event archives.

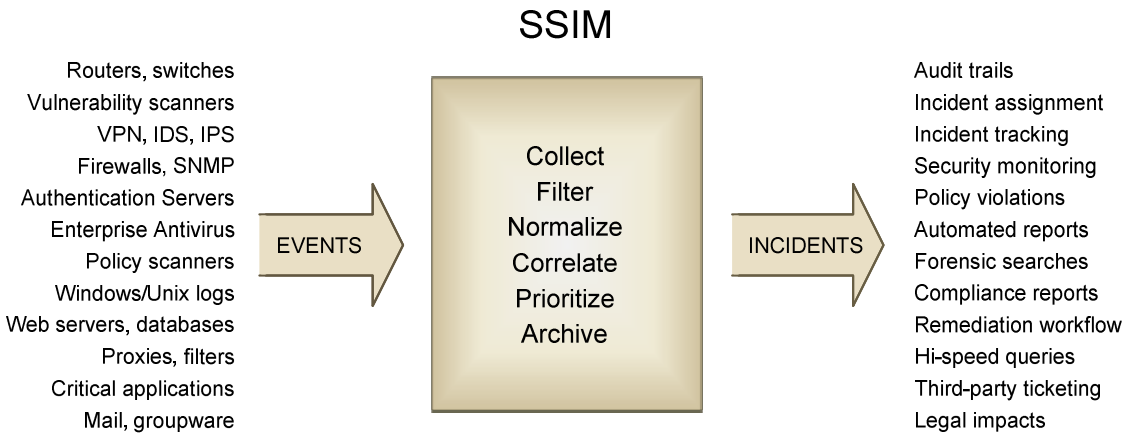

Figure 1 – Symantec Security Information Manager Workflow

[1] SQL – Structured Query Language

The SSIM solution consists of the following major components:

- **Information Manager** – comprises the core functionality of a SSIM deployment. It installs on a standard server platform that supports RHEL (Red Had Enterprise Linux). It aggregates and processes event data for correlation, incident management, and archival. It also consists of a comprehensive report engine and an embedded LDAP (Lightweight Directory Access Protocol) directory for centralized access control and multi-domain management. The various roles of the Information Manager (Collection, Correlation, Archival, and Service Provider) can be distributed in local clusters and then federated for fail safety, higher scalability, and global deployments.
- **Event Collector** – gathers and filters events from event sources. Collectors are installed directly on a security point product or in strategic locations with access to security events. They use application specific *sensors* that retrieve events from a file, database, or syslog. SSIM provides multi-vendor support with over 200 predefined collectors for popular products. It also has universal collectors that can be customized for unique event sources.
- **Event Agent** – manages the secure communication path between collectors and the Information Manager. This path is used to remotely configure collectors and sensors, and to forward both raw event data and events that have been filtered and aggregated. The Event Agent is a Java-based application that is always installed on the same computer or device as the collector component.
- **Information Manager Console** – provides a bidirectional administrative GUI[2] to the Information Manager. It is a Java-based application used by administrators, analysts, and service desk systems to perform security monitoring functions, such as incident management, reports, and rule definition. It presents both high-level and detailed views of critical security information.
- **Web Configuration Interface** and **OpenSSH –** provide administrative access to the Information Manager server via a web browser or a remote secure shell (SSH) connection.

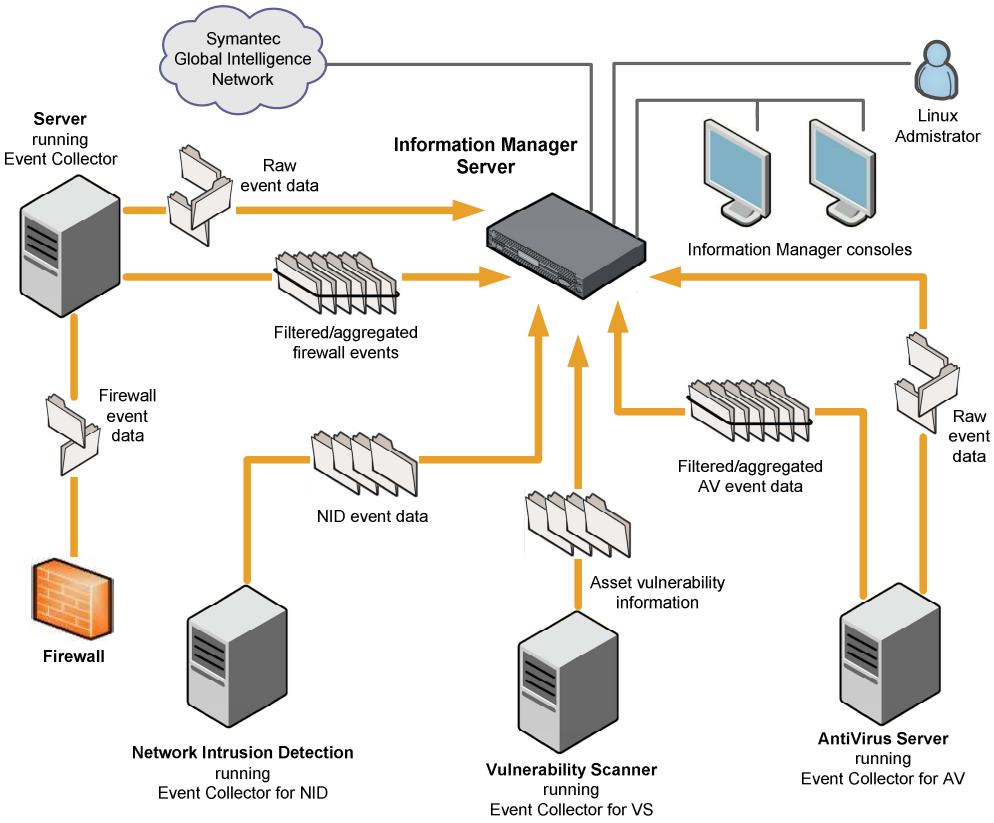

Figure 2 – SSIM Architecture Overview

[2] GUI – Graphical User Interface

2.1.2 Symantec Cryptographic Module Security

The Symantec Cryptographic Module, SymCrypt, is a software shared library that resides on various Symantec application components, including the Information Manager server in SSIM. It provides the primitive cryptographic services required by SSH and TLS[3] for secure communication. The module includes implementations of the following FIPS-Approved algorithms:

- Advanced Encryption Standard (AES)
- Triple Data Encryption Algorithm (TDEA or Triple-DES[4])
- Secure Hash Algorithm (SHA)
- (Keyed-) Hash Message Authentication Code (HMAC)
- Digital Signature Algorithm (DSA)
- RSA[5] signature generation and verification
- ANSI[6] X9.31 Pseudo Random Number Generator (PRNG)

The Symantec Cryptographic Module operates in either a FIPS-Approved mode of operation or a non-FIPS mode of operation. It is validated at the following FIPS 140-2 Section levels:

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	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[7]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The Symantec Cryptographic Module is a software module with a multi-chip standalone embodiment. The overall security level of the module is 1. SymCrypt is implemented in the C programming language and consists of a shared library that is linked with SSIM application components. It is designed to execute on a host system with a General Purpose Computer (GPC) hardware platform. The following sections define the physical and logical boundary of the SymCrypt module.

[3] TLS – Transport Layer Security
[4] DES – Data Encryption Standard
[5] RSA – Rivest, Shamir, Adleman
[6] ANSI - American National Standards Institute
[7] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, there are no physical protection mechanisms implemented. Therefore, the module must rely on the physical characteristics of the host system. The physical boundary of the cryptographic module is defined by the hard enclosure around the host system on which it runs. The module supports the physical interfaces of a GPC, including the integrated circuits of the system board, the CPU, network adapters, RAM, hard disk, device case, power supply, and fans. Other devices may be attached to the GPC, such as a display monitor, keyboard, mouse, printer, or storage media. See Figure 3 below for a standard GPC block diagram.

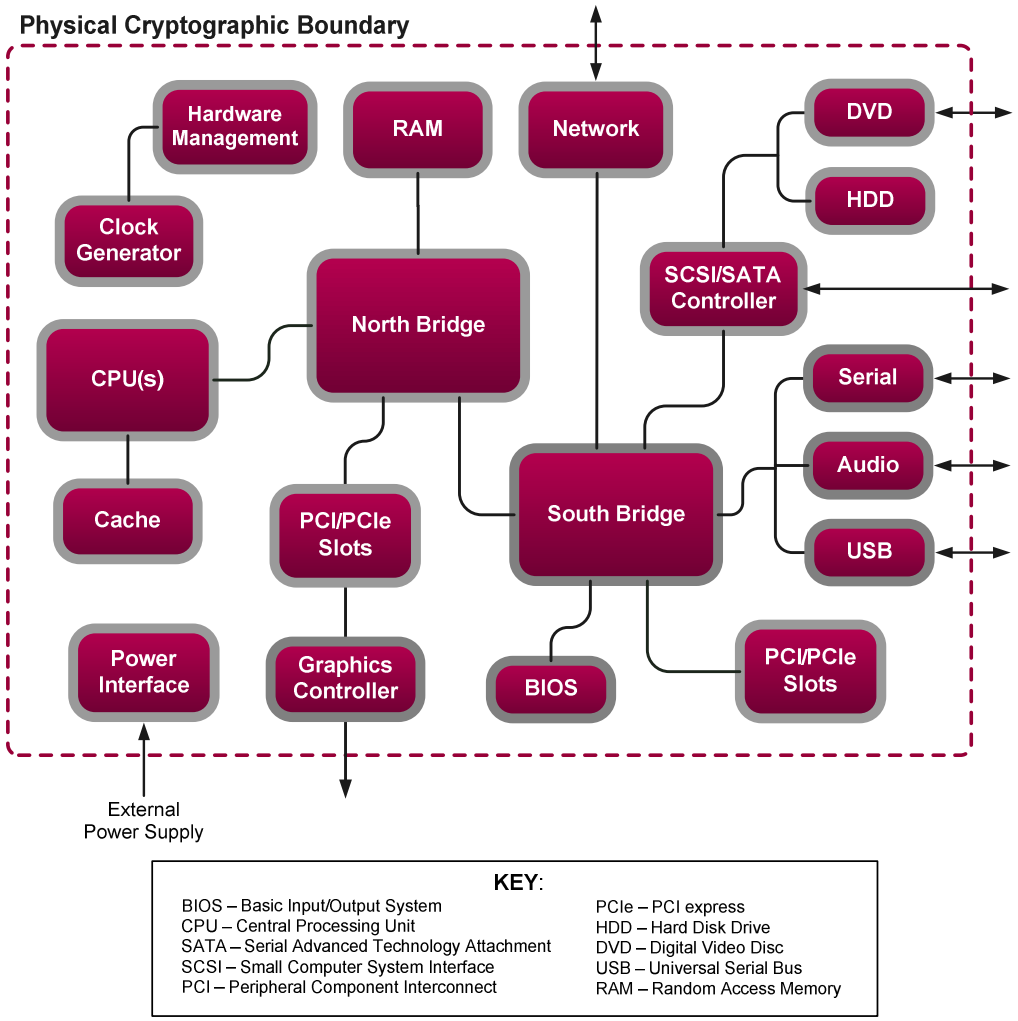

Figure 3 – Standard GPC Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 4 shows a logical block diagram of the module executing in memory and its interactions with surrounding software components, as well as the module's logical cryptographic boundary. The module's services are designed to be called by other Symantec software components.

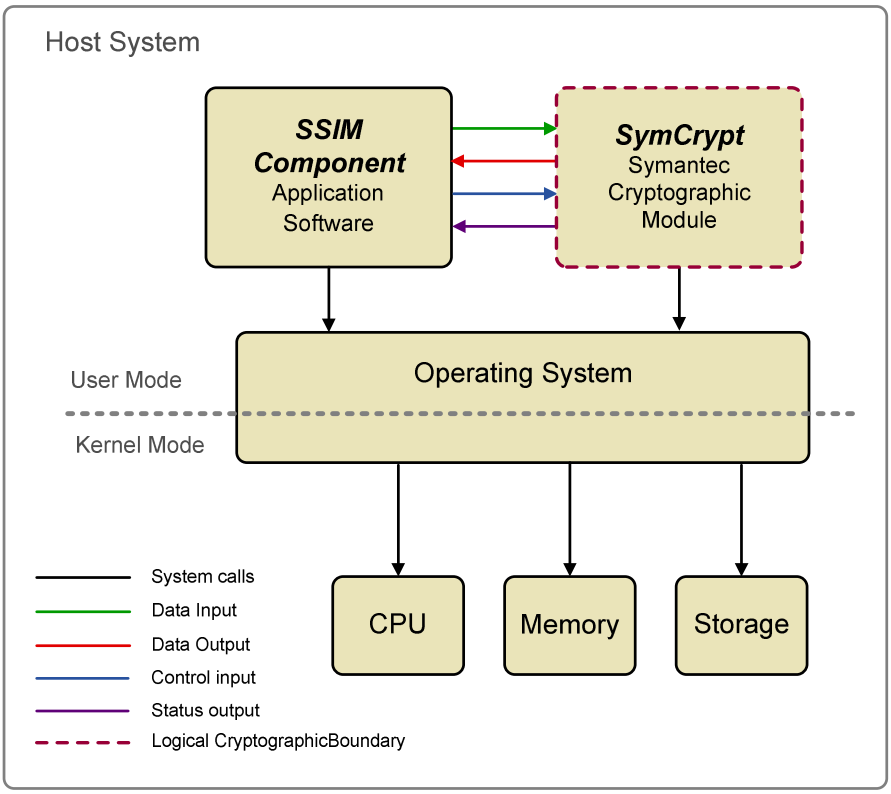

Figure 4 – Logical Block Diagram and Cryptographic Boundary

2.3 Module Interfaces

The module's logical interfaces exist at a low level in the software as an Application Programming Interface (API). Both the API and physical interfaces can be categorized into following interfaces defined by FIPS 140-2: Data Input, Data Output, Control Input, Status Output, Power Input.

A mapping of the FIPS 140-2 logical interfaces, the physical interfaces, and the module interfaces can be found in the following table:

Table 2 – FIPS 140-2 Interface Mappings

FIPS Interface	Physical Interface	Module Interface (API)
Data Input	USB ports (keyboard, mouse, data), network ports, serial ports, SCSI/SATA ports, DVD drive	Arguments for API calls that contain data to be used or processed by the module
Data Output	Monitor, USB ports, network ports, serial ports, SCSI/SATA ports, audio ports, DVD drive	Arguments for API calls that contain or point to where the result of the function is stored
Control Input	USB ports (keyboard, mouse), network ports, serial ports, power switch	API Function calls and parameters that initiate and control the operation of the module

FIPS Interface	Physical Interface	Module Interface (API)
Status Output	Monitor, serial ports, network ports	Return values from API function calls and error messages
Power Input	Power Interface	N/A

2.4 Roles and Services

The Symantec Cryptographic Module supports the following two roles for operators, as required by FIPS 140-2: Crypto-Officer (CO) role and User role. As allowed by FIPS 140-2, the module does not perform authentication of any operators. Both roles are implicitly assumed when services are executed.

Note 1: Table 3 and Table 4 use the following definitions for "CSP[8] and Type of Access".

R – Read: *The plaintext CSP is read by the service.*
W – Write: *The CSP is established, generated, modified, or zeroized by the service.*
X – Execute: *The CSP is used within an Approved (or allowed) security function or authentication mechanism.*

Note 2: Input parameters of an API call that are not specifically a signature, hash, message, plaintext, ciphertext, or a key are NOT itemized in the "Input" column, since it is assumed that most API calls will have such parameters.

Note 3: The "Input" and "Output" columns are with respect to the module's logical boundary.

2.4.1 Crypto Officer Role

The operator in the Crypto Officer role installs, uninstalls, and administers the module via the host platform's OS interfaces. An operator assumes the CO role by invoking one of the following services:

Table 3 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Initialize FIPS mode	Performs integrity check and power-up self-tests. Sets the FIPS mode flag to on	API call parameters	Status	Integrity check HMAC key, ANSI X9.31 PRNG seed, ANSI X9.31 PRNG seed key
Exit FIPS mode	Sets the FIPS mode flag to off	API call parameters	Status	None
Show status	Returns the current mode of the module (FIPS or non-FIPS)	None	Status	None
Run self-tests on demand	Performs power-up self-tests	None	Status	Integrity check HMAC key

2.4.2 User Role

The operator in the User role is a consumer of the module's security services. The role is assumed by invoking one of the following cryptographic services:

[8] CSP – Critical Security Parameter

Table 4 – User Services

Service	Description	Input	Output	CSP and Type of Access
Generate random number (ANSI X9.31)	Returns the specified number of random bits to calling application	API call parameters	Status, random bits	ANSI X9.31 RNG seed – RWX ANSI X9.31 seed key – RX
Generate message digest (SHS[9])	Compute and return a message digest using SHS algorithms	API call parameters, message	Status, hash	None
Generate keyed hash (HMAC)	Compute and return a message authentication code using HMAC-SHAx	API call parameters, key, message	Status, hash	HMAC key – RX
Zeroize key	Zeroizes and de-allocates memory containing sensitive data	API call parameters	Status	AES key – W TDES key – W HMAC key – W RSA private/public key – W DSA private/public key – W DH[10] components – W
Symmetric encryption	Encrypt plaintext using supplied key and algorithm specification (TDES or AES)	API call parameters, key, plaintext	Status, ciphertext	AES key – RX TDES key – RX
Symmetric decryption	Decrypt ciphertext using supplied key and algorithm specification (TDES or AES)	API call parameters, key, ciphertext	Status, plaintext	AES key – RX TDES key – RX
Generate asymmetric key pair	Generate and return the specified type of asymmetric key pair (RSA or DSA)	API call parameters	Status, key pair	RSA private/public key – W DSA private/public key – W
RSA encryption	Encrypt plaintext using RSA public key (used for key transport)	API call parameters, key, plaintext	Status, ciphertext	RSA public key – RX
RSA decryption	Decrypt ciphertext using RSA private key (used for key transport)	API call parameters, key, ciphertext	Status, plaintext	RSA private key – RX
DH key agreement	Perform key agreement using Diffie-Hellman algorithm	API call parameter	Status, key components	DH components – W
Signature Generation	Generate a signature for the supplied message using the specified key and algorithm (RSA or DSA)	API call parameters, key, message	Status, signature	RSA private key – RX, DSA private key – RX
Signature Verification	Verify the signature on the supplied message using the specified key and algorithm (RSA or DSA)	API call parameters, key, signature, message	Status	RSA public key – RX DSA public key – RX

[9] SHS – Secure Hash Standard
[10] DH – Diffie-Hellman

2.5 Physical Security

The Symantec Cryptographic Module is a software module and does not include physical security mechanisms. Thus, the FIPS 140-2 requirements for physical security are not applicable.

2.6 Operational Environment

The module was tested and found to be compliant with FIPS 140-2 requirements on the following platforms:

- GPC with an Intel Pentium 4 processor running Windows 2003 Server 32-bit
- GPC with an Intel Xeon x3430 processor running Red Hat Enterprise Linux (RHEL) 4.8 32-bit.

Symantec affirms that the module also executes in its FIPS-Approved manner (as described in this Security Policy) on other Operating Systems that are binary-compatible to those on which the module was tested. All cryptographic keys and CSPs are under the control of the operating system, which protects the CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to CSPs through its well-defined API.

2.7 Cryptographic Key Management

The module implements the following FIPS-Approved algorithms:

Table 5 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
AES in ECB[11], CBC[12], OFB[13], CFB[14]8, CFB128 modes with 128, 192, and 256 bit keys	1607
Triple-DES in ECB, CBC, CFB8, CFB64, OFB modes with 112 and 168 bit keys	1052
RSA (ANSI X9.31, PKCS[15] #1.5, PSS) sign/verify with 1024, 1536, 2048, 3072, 4096 bit keys	789
RSA (ANSI X9.31) key generation with 1024, 1536, 2048, 3072, 4096 bit keys	789
DSA sign/verify and key generation with 1024 bit keys	498
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1420
HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	943
ANSI X9.31PRNG Appendix A.2.4	861

The module utilizes the following non-Approved algorithms, which are allowed for use in a FIPS-Approved mode of operation:

- RSA key wrapping (1024- to 15360-bit keys)
    - Key establishment methodology provides between 80 and 256 bits of encryption strength

[11] ECB – Electronic Codebook
[12] CBC – Cipher Block Chaining
[13] OFB – Output Feedback
[14] CFB – Cipher Feedback
[15] PKCS – Public-Key Cryptography Standards

- Diffie-Hellman key agreement (1024-bit key)
  - Key establishment methodology provides 80 bits of security
- Message Digest 5 (MD5)
  - Message authentication for use within the TLS Key Derivation Function (KDF)

Additionally, the module implements the following non-Approved algorithms, which are only available in a non-Approved mode of operation:

- DES
- Blowfish
- CAST (Carlisle Adams and Stafford Tavares)
- IDEA (International Data Encryption Algorithm)
- RC2 (Rivest Cipher 2)
- RC4
- RC5
- MD2
- MD4
- RipeMD (RACE[16] Integrity Primitives Evaluation MD)
- MDC-2 (Modification Detection Code 2)

The CSPs supported by the module are shown in Table 6 below.

Note: The "Input" and "Output" columns in Table 6 are in reference to the module's logical boundary. Keys that enter and exit the module via an API call parameter are in plaintext.

Table 6 – List of Cryptographic Keys, Key Components, and CSPs

CSP/Key	CSP/Key Type	Input	Output	Storage	Zeroization	Use
HMAC key (Integrity check)	HMAC key	Never	Never	In module binary	By uninstalling the module	Software integrity check
AES key	AES128, 192, 256 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Encryption, decryption
TDES key	TDES 112, 168 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Encryption, decryption
HMAC key	HMAC key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Message Authentication with SHS
RSA private key	RSA 1024, 1536, 2048, 3072, 4096 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature generation, decryption
		Internally generated	API call parameter			Used by host application
RSA public key	RSA 1024, 1536, 2048, 3072, 4096 bit	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature verification, encryption

[16] RACE – Research and Development in Advanced Communications Technologies in Europe

	key	Internally generated	API call parameter			Used by host application
DSA private key	DSA 1024 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature generation
		Internally generated	API call parameter			Used by host application
DSA public key	DSA 1024 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature verification
		Internally generated	API call parameter			Used by host application
DH public components	Public components of DH protocol	Internally generated	API call parameter	Plaintext in volatile memory	By API call, power cycle, host reboot	Used by host application
DH private component	Private exponent of DH protocol	Internally generated	API call parameter	Plaintext in volatile memory	By API call, power cycle, host reboot	Used by host application
ANSI X9.31 PRNG seed	128 bit random value	Internally generated	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Generate random number
ANSI X9.31 PRNG seed key	AES 256 bit key	Internally generated	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Generate random number

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 PRNG implementation to generate cryptographic keys. This PRNG is FIPS-Approved as shown in Annex C to FIPS PUB 140-2.

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output from its physical boundary. However, keys are passed to the module as parameters from the applications resident on the host platform via the exposed APIs. Similarly, keys and CSPs exit the module in plaintext via the well-defined exported APIs.

2.7.3 Key/CSP Storage and Zeroization

Symmetric, asymmetric, and HMAC keys are either provided by or delivered to the calling process, and are subsequently destroyed by the module at the completion of the API call. Keys and CSPs stored in RAM can be zeroized by a power cycle or a host system reboot. The X9.31 PRNG seed and seed key are initialized by the module at power-up and remain stored in RAM until the module is uninitialized by a host system reboot or power cycle. The HMAC key that is used to verify the integrity of the module is hard-coded within the module binary.

2.8 EMI/EMC

SymCrypt is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which SymCrypt resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal

Regulations Part 15. However, all systems sold in the United States must meet these applicable FCC requirements.

2.9 Self-Tests

2.9.1 Power-Up Self-Tests

The Symantec Cryptographic Module performs the following self-tests at power-up:

- Software integrity check
  - This test calculates a HMAC SHA-1 digest of the module and compares it to the pre-calculated digest stored in the module's associated digest file.
- Known Answer Tests (KATs)
  - SHA-1
  - HMAC-SHA1
  - HMAC-SHA-224
  - HMAC-SHA-256
  - HMAC-SHA-384
  - HMAC-SHA-512
  - Triple-DES-ECB 112 and 168 bit key encrypt/decrypt
  - AES-ECB 128 bit key encrypt/decrypt
  - ANSI X9.31 PRNG
  - RSA for sign/verify
- DSA pairwise consistency

If a power-up self-test fails, the module will enter an error state, during which cryptographic functionality and all data output is inhibited. To clear the error state, the CO must reinitialize the module.

2.9.2 Conditional Self-Tests

The Symantec Cryptographic Module performs the following conditional self-tests:

- Continuous RNG test
- RSA pairwise consistency for sign/verify and encrypt/decrypt
- DSA pairwise consistency

If a conditional self-test fails, the module will enter an error state, during which cryptographic functionality and all data output is inhibited. To clear the error state, the CO must reinitialize the module.

2.10 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3 Secure Operation

The Symantec Cryptographic Module meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

3.1 Initial Setup

FIPS 140-2 mandates that a software cryptographic module at Security Level 1 be restricted to a single operator mode of operation. Prior to installing the module, the Crypto-Officer must ensure the host system OS is configured for single-user mode. The CO must disable any unnecessary guest accounts in order to ensure that only an authorized operator can log into the OS. The following Windows services should be carefully evaluated by the CO and turned off if applicable:

- Fast-user switching (irrelevant if server is a domain member)
- Terminal services
- Remote registry service
- Secondary logon service
- Telnet service
- Remote desktop and remote assistance service
- Network Information Service (NIS) and other name services for users and groups.

The SymCrypt module is installed as part of the installation of a Symantec product. For SSIM, the CO should follow the installation procedures found in *Symantec Security Information Manager Installation Guide*. These instructions install both server and client components. The Information Manager which is installed on a server platform comes with RHEL 4.8 already configured for single-user mode.

3.2 Crypto–Officer Guidance

The Crypto-Officer is required to install and initialize the module to run in a FIPS mode of operation. When the module is loaded by the host SSIM software application, an internal global flag *fips_mode* is set to FALSE. At this point the module is considered to be uninitialized in non-FIPS mode. A single initialization function call, *FIPS_mode_set (onoff)* with a non-zero *onoff*, is required to operate the module in a FIPS-Approved mode of operation. This function checks the integrity of the module using an HMAC-SHA-1 digest. If the integrity check succeeds, then the module performs power-up self-tests. If the module passes all self-tests, the global flag *fips_mode* is set to TRUE and the function returns a value of “1”, which indicates the module is in a FIPS-Approved mode of operation.

Self-tests can be performed on demand by cycling the power on the host device, or by the function call *FIPS_selftest()*.

3.3 User Guidance

The SymCrypt module is designed for use by Symantec software applications. SymCrypt does not input, output, or persistently store CSPs with respect to the physical boundary. The User (Symantec software component, in this case) is responsible for providing persistent storage of the cryptographic keys and CSPs, and to ensure that keys are transmitted outside the physical cryptographic boundary in a secure manner. Only security functions that are FIPS-Approved or allowed for use in the FIPS mode of operation are available when configured for FIPS mode. Should an application attempt to use a non-Approved security function, SymCrypt will give an error message stating the function is non-Approved and will lock the module from further use. SymCrypt must then be reinitialized by the Crypto-Officer.

4 Acronyms

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CAST	Carlisle Adams and Stafford Tavares
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CTR	Counter
DAS	Direct-Attached Storage
DES	Data Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECB	Electronic Code Book
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GIN	Global Information Network
GPC	General Purpose Computer
GUI	Graphical User Interface
HMAC	(Keyed-) Hash Message Authentication Code
IDEA	International Data Encryption Algorithm
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
KAT	Known Answer Test
KDF	Key Derivation Function
LDAP	Lightweight Directory Access Protocol

Acronym	Definition
MAC	Message Authentication Code
MD	Message Digest
MDC	Modification Detection Code
NAS	Network-Attached Storage
NIS	Network Information Service
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
OS	Operating System
PKCS	Public-Key Cryptography Standards
PRNG	Pseudo Random Number Generator
PSS	Probabilistic Signature Scheme
RACE	Research and Development in Advanced Communications Technologies in Europe
RC	Rivest Cipher
RHEL	Red Hat Enterprise Linux
RipeMD	RACE Integrity Primitives Evaluation Message Digest
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman
SAN	Storage Area Network
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SNMP	Simple Network Management Protocol
SOC	Security Operations Center
SP	Special Publication
SQL	Structured Query Language
SSH	Secure Shell
SSIM	Symantec Security Information Manager
TDEA	Triple Data Encryption Algorithm
TLS	Transport Layer Security
USB	Universal Serial Bus
VPN	Virtual Private Network

Prepared by:
Corsec Security, Inc.

10340 Democracy Lane, Suite 201
Fairfax, VA 22030

Phone: (703) 267-6050
Email: info@corsec.com
http://www.corsec.com