Symantec Cryptographic Module 1.0

FIPS 140-2 Non-Proprietary Security Policy 1.0

Symantec Cryptographic Module

Documentation version 1.0

Contents

Symantec Cryptographic Module 1.0 Security Policy

Introduction

This document is the non-proprietary security policy for the Symantec Cryptographic Module. It was submitted as part of the Federal Information Processing Standard (FIPS) 140-2 Level 1 validation process.

For more information about the standards and the validation process for cryptographic modules, visit the National Institute of Standards and Technology (NIST) Computer Security Resource Center (CSRC) Web site at the following Web address:

http://csrc.nist.gov/cryptval/

Cryptographic Module

The Symantec Cryptographic Module is a software library that contains only FIPS-approved cryptographic algorithms. It can be executed on computer systems that run Microsoft Windows 98, Me, NT, 2000, XP, and 2003, but it has only been validated on computer systems that are running Windows 2000 and Windows XP Professional.

For the purposes of FIPS 140-2 Level 1 validation, version 1.0 of the Symantec Cryptographic Module library (hereafter called the SymCrypt library) is a single, dynamic link library (DLL) named SymCrypt.dll. It provides a C-language application program interface (API) for use by other processes that require message digest and symmetric encryption and decryption functionality.

The SymCrypt library is classified as a *multi-chip standalone module*. The *logical cryptographic boundary* of the SymCrypt library is the DLL file itself. The *physical boundary* of the SymCrypt library is the enclosure of the computer system on which it is executing. The SymCrypt library does not communicate with anything other than the process that calls it. It makes no network or inter-process connections and creates no files.

The SymCrypt library was tested on a DELL OptiPlex GX1 computer running Microsoft Windows XP Professional and on a DELL OptiPlex GX1 computer running Microsoft Windows 2000. The system meets level 3 EMI/EMC security requirements - Subpart B of FCC part 15 requirements. (FCC Certificate Proprietary & Confidential Number JGY-INF10280).

Table 1-1 lists the validation levels for each FIPS 140-2 section.

Table 1-1 FIPS 140-2 Validation Levels

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	Not applicable
6	Operational Environment	1
7	Crytographic Key Management	1
8	EMI/EMC	3
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	Not applicable
14	Cryptographic Module Security Policy	1

Roles and Services

The SymCrypt library supports the following authorized operator roles:

- User Role: The *User* is any entity that can load the SymCrypt library DLL and call any of its API functions.
  The User role has access to all of the services that are provided by the SymCrypt library.
- Crypto Officer Role: The *Crypto Officer* is any entity that can install the SymCrypt library on the operating system.
  The Crypto Officer role is assumed implicitly when the operator of the computer installs the SymCrypt library.

The SymCrypt library does not supply a Maintenance role or mode.

The SymCrypt library implements the following FIPS-approved algorithms:

Windows XP Professional

- SHA-1 hashing algorithm [Cert #248] (FIPS 180-2)
- HMAC-SHA-1 keyed hash message authentication [Cert #5] (FIPS 198a)
- AES symmetric cipher algorithm in CBC or ECB mode with key lengths of 128, 192, or 256 bits [Cert #164] (FIPS 197)
- Triple-DES (TDES) symmetric cipher algorithm in ECB mode with a key length of 168 bits [Cert #266] (FIPS 46-3)
- ANSI X9.31 random number generation algorithm [Cert #12]

Windows 2000 Professional

- SHA-1 hashing algorithm [Cert #248] (FIPS 180-2)
- HMAC-SHA-1 keyed hash message authentication [Cert #5] (FIPS 198a)
- AES symmetric cipher algorithm in CBC or ECB mode with key lengths of 128, 192, or 256 bits [Cert #164] (FIPS 197)
- Triple-DES (TDES) symmetric cipher algorithm in ECB mode with a key length of 168 bits [Cert #266] (FIPS 46-3)
- ANSI X9.31 random number generation algorithm [Cert #12]

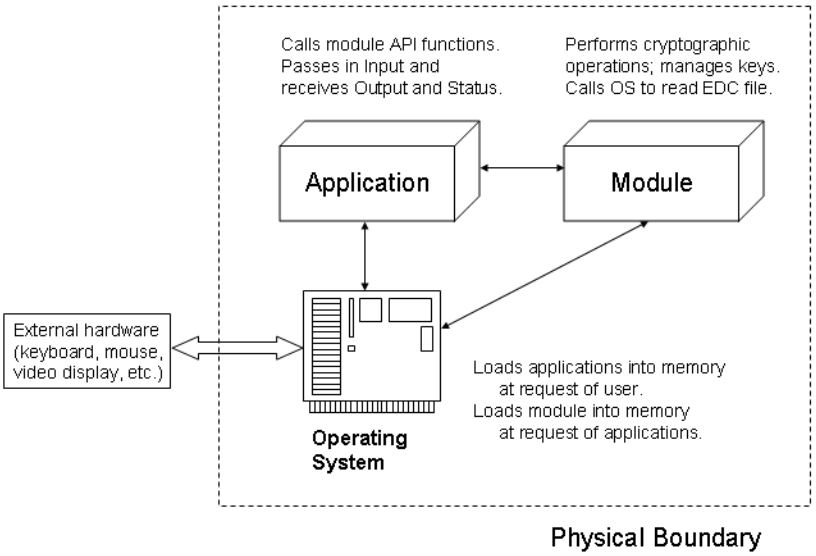
API functions exist to provide other processes with access to the SHA-1, TDES, and AES algorithms. No external API exists to access the ANSI X9.31 random number generation algorithm. It is used internally to create random AES key data and random TDES key data.

Ports and Interfaces

The *physical ports* of the SymCrypt library are the same as the computer system on which it is executing. The *logical interface* is a C-language application program interface (API).

Figure 1-1 shows the physical boundary of the Symantec Cyptographic Module.

Figure 1-1 Symantec Crytographic Module Block Diagram



The SymCrypt library maintains separation of the Data Input, Data Output, Control Input, and Status Output interfaces. The Data Input interface contains the input parameters of the API functions. The Data Output interface contains the output parameters of the API functions. The Control Input interface contains the actual API functions. The Status Output interface includes the return values of the API functions and the output parameters of the ScmGetStatus() function.

Table 1-2 lists the types of data that are received and sent through each interface.

Table 1-2 SymCrypt Interface Data Types

Interface	Data Type
Data Input	Key Data
	Key Parameters
	Plaintext for encryption
	Ciphertext for decryption
	Plaintext for hashing
Data Output	Key Data
	Key Parameters
	Ciphertext from the encryption process
	Plaintext from the decryption process
	Hash value
Control Input	Initialize/Shutdown
	Key Generation
	Encryption/Decryption start
	Destroy key data
	Create/Destroy hash
	Self-test
Status Output	Error returned by most recent API function call

Table 1-3 lists the functions that comprise the logical interface of the SymCrypt library API. Processes that require FIPS 140-2 approved cryptography can use all of these functions because the SymCrypt library does not contain any non-approved algorithms or modes.

Table 1-3 SymCrypt API Functions

Service	Function
Setup	ScmInitialize
Shutdown	ScmShutdown
AES and TDES Key Creation	ScmImportKey
	ScmSetKeyInfo

Table 1-3 SymCrypt API Functions

Service	Function
AES and TDES Key Data Export	ScmExportKey ScmGetKeyInfo
AES and TDES Encryption	ScmBeginEncrypt ScmEncrypt ScmFinishEncrypt
AES and TDES Decryption	ScmBeginDecrypt ScmDecrypt ScmFinishDecrypt
Destruction and Zeroization of key data in memory	ScmDestroyKey ScmClearAllData
SHA-1 Hashing	ScmCreateHash ScmHashData ScmHashValue ScmDestroyHash **Note:** The API for SHA-1 Hashing does not allow the data length to be zero (0).
Self-Test	ScmPerformSelfTest
Status	ScmGetStatus

Finite State Model

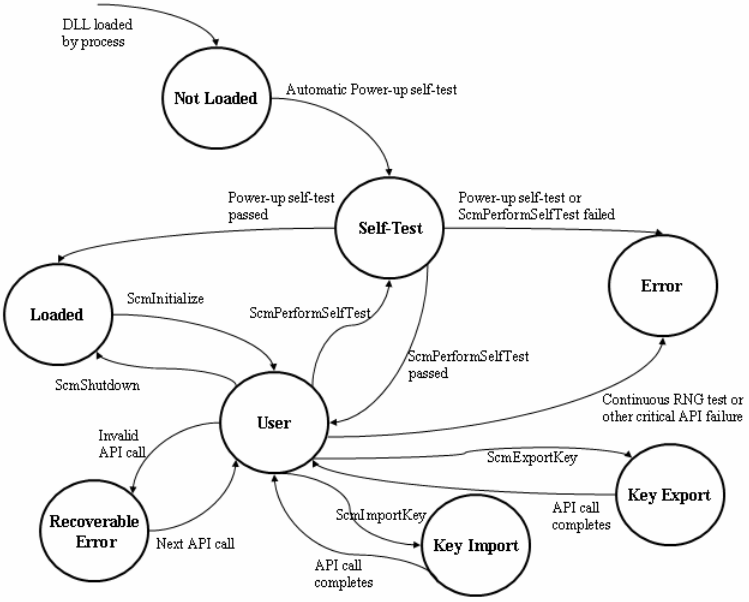
The SymCrypt library maintains a current state value that can be read through the ScmGetStatus() API function. Table 1-4 identifies the various states and how they are entered.

Table 1-4 State Table

State	Purpose	Valid API Functions	State Change
Not Loaded	Not applicable.	None	When loaded, go to Self-Test state
Self-Test	Verify integrity of module and algorithms.	ScmGetStatus()	If any test fails, go to unrecoverable Error state. If all tests succeed, return to the previous state. If previous state was Not Loaded, go to Loaded state.
Error	Notify user that a fault has occurred.	ScmGetStatus()	If error is unrecoverable, module must be unloaded. If error is recoverable, the next function call will clear the error.
Loaded	Module has no key data, and no operator role is active.	ScmGetStatus(), ScmInitialize(), ScmPerformSelfTest()	Call ScmInitialize() to move the module into User state. Call ScmPerformSelfTest() to move the module into Self-Test state.
User	Module has the User operator role active, can accept data and control input, and can return data output.	All functions	Call ScmShutdown() to return the Module to Loaded state. Call ScmPerformSelfTest() to move the module to Self-Test state.
Key Import	Operator has called the ScmImportKey function. The Module reads the key data from the data input port.	Not applicable	Module returns to User state upon completion of the ScmImportKey API function.
Key Export	Operator has called the ScmExportKey function. The Module writes the key data to the data output port.	Not applicable	Module returns to User state upon completion of the ScmExportKey API function.

Figure 1-2 shows each state and the transition between states.

Figure 1-2 SymCrypt Library State Transition Diagram



Operational Environment

The Microsoft Windows operating system segregates user processes into process spaces. Each process space is an independent virtual memory area that is logically separated from all other process by the operating system and CPU. The SymCrypt library functions completely within the process space of the process that loads it. The SymCrypt library does not communicate with any processes other than the one that loads it, and satisfies the FIPS 140-2 requirement for a single user mode of operation.

Cryptographic Key Management

Only the process that created or imported an AES or TDES key can export or use it. This is enforced both by the operating system and by additional checks in the SymCrypt library. No persistent storage of key data is performed. All API functions in the SymCrypt library are executed in mutual exclusion such that no two API functions will execute at the same time.

Table 1-5 lists the types of secure data that is managed by the SymCrypt library.

Table 1-5 Key Management Mechanisms

Role	Service	Key and CSP Data	Access Rights by Operators
User	Setup	None	Not applicable
User	Shutdown	None	Not applicable
User	SHA-1 Hashing	None	Not applicable
User	Status	None	Not applicable
User	Self-Test	HMAC-SHA-1 secret key	Use
User	AES and TDES Key Data Import	Secret keys	Write
User	AES and TDES Key Data Export	Secret keys	Read
User	AES and TDES Encryption	Secret keys	Use
User	AES and TDES Decryption	Secret keys	Use
User	Destruction and Zeroization of key data in memory	Secret keys	Write

A user can perform key zeroization of a single AES or TDES key, or of all AES and TDES keys, by calling an API function. When the DLL is unloaded from memory, all AES keys, all TDES keys, and the random number generator's seed and seed key are zeroized.

The HMAC-SHA-1 exists in the module's DLL file and can be zeroized by removing the file's image from the hard drive upon which is installed.

Self-Tests

The SymCrypt library conducts module integrity and algorithm self-tests to guarantee correct operation. The tests are performed automatically when the SymCrypt library is loaded, without the need for operator input. Self-tests are also performed when the self-test API function is called. If any self-test fails, the SymCrypt library enters an unrecoverable error state and prevents itself from being used for any cryptographic purpose.

The power-up self-tests are as follows:

- SHA-1 known answer test
- HMAC-SHA-1 known answer test
- AES known answer test
- TDES known answer test
- Random number generator known answer test
- DLL file integrity test

The known answer tests function by providing specific input to the algorithm and testing the output against a specific expected value. The DLL file integrity test computes an HMAC-SHA-1 value of the DLL file's contents, loads a file named SymCrypt.edc that contains the expected HMAC-SHA-1 result, and compares the two values. If they do not match, or if the SymCrypt.edc file does not exist, the module integrity self-test fails, and the SymCrypt library refuses access to all of its services.

The random number generator is also tested on a continual basis. All newly generated random data is compared with the previously generated data. If a match is found, the continuous random number generator self-test fails.

Mitigation of Other Attacks

The SymCrypt library does not contain additional security mechanisms besides the requirements for FIPS 140-2 Level 1 cryptographic modules.