Cimcor Cryptographic Module

FIPS 140-2 Level 2 Security Policy

Version: 1.7
Last Updated: March 25, 2010

CIMCOR, Inc.
Corporate Headquarters
8252 Virginia Street, Suite C
Merrillville, IN 46410
Toll Free: 877-424-6267
Intl: (219) 736-4400
Fax: (219) 736-4401

Revision History

Authors	Date	Version	Comment
Cimcor, Inc.	2009-02-08	1.0	Initial draft
Cimcor, Inc.	2009-06-20	1.1	New label for block diagram in section 1. Two Certificates columns for Algorithms table in section 5, removed D-H. References to both level 1 and level 2 as appropriate.
Cimcor, Inc.	2009-06-24	1.2	2.0: designate platforms 2.1: specify missing platforms 4.1: renumber 5.0: drop superfluous column
Cimcor, Inc.	2009-06-26	1.3	3.1: remove unnecessary reference to PINs and passwords, refer to D-H primitives.
Cimcor, Inc.	2009-07-16	1.4	New block diagram
Cimcor, Inc.	2010-02-02	1.5	Updated for comments, all supported operational environments
Cimcor, Inc.	2010-03-12	1.6	4.1: verification at installation
Cimcor, Inc.	2010-03-25	1.7	Tested platforms listing for consistency

Table of Contents

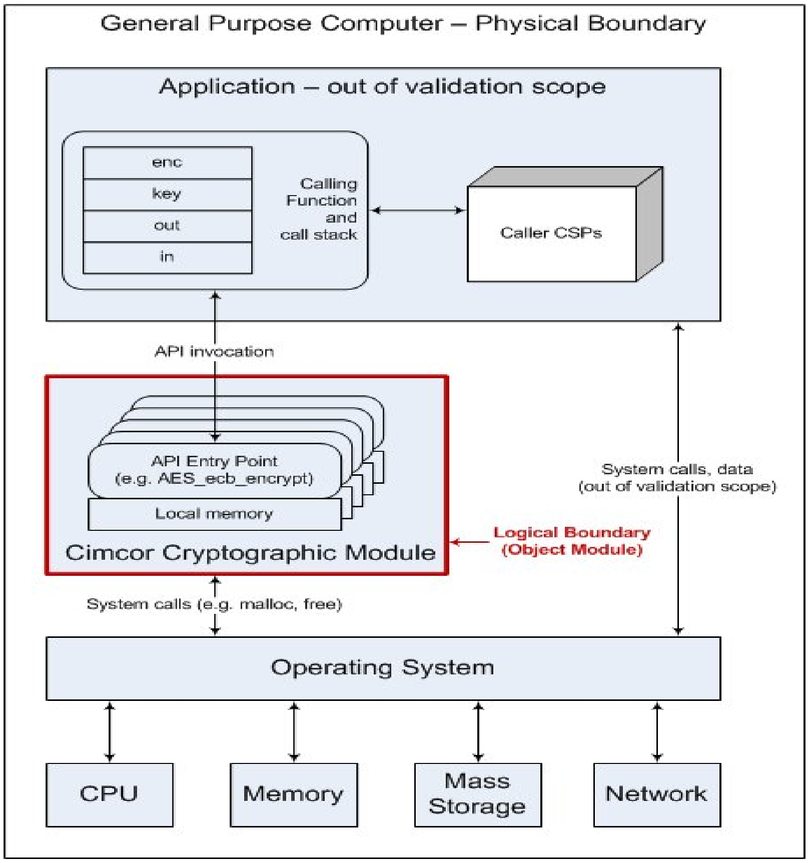
1. Introduction

This document comprises the non-proprietary FIPS 140-2 Security Policy for the Cimcor Cryptographic Module (Software Version 1.0).

FIPS 140-2, *Security Requirements for Cryptographic Modules*, specifies the requirements for cryptographic modules. For more information about the FIPS 140-2 standard and the cryptographic module validation process see http://csrc.nist.gov/groups/STM/cmvp/index.html.

2. Cimcor Cryptographic Module Overview

The Cimcor Cryptographic Module (Module) is a software library that provides symmetric and asymmetric encryption, hashing, and random number generation functions on a wide variety of computing platforms. The Module is the specific shared library file with the name given in the *Supported Platforms* table in section 2.1 below.


General Purpose Computer – Physical Boundary
Application – out of validation scope
enc
key
out
in
Calling Function and call stack
Caller CSPs
API invocation
API Entry Point (e.g. AES_ecb_encrypt)
Local memory
Cimcor Cryptographic Module
Logical Boundary (Object Module)
System calls, data (out of validation scope)
System calls (e.g. malloc, free)
Operating System
CPU
Memory
Mass Storage
Network

Block Diagram

For FIPS 140-2 validation purposes the Module is a multi-chip embedded module. The logical cryptographic boundary of the Module is the shared library file. The Module exchanges data only with the calling application, and does not perform any network or inter-process communication and does not read or write any persistent storage.

The Module meets the requirements applicable to FIPS 140-2 Level 2. The Module relies on the host operating system for authentication of operators and protecting and clearing authentication data when the computer is powered down or restarted. The operating systems hosting the Module enforce identity-based authentication of roles prior to operator access to Module functions.

Requirement	Compliance Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services, and Authentication	3
Finite State Model	2
Physical Security	N/A
Operational Environment	2
Cryptographic Key Management	2
Electromagnetic Interference/Electromagnetic Compatibility	2
Self-Tests	2
Design Assurance	2
Mitigation of Other Attacks	N/A
Overall	2

Compliance

2.1 Supported Platforms

The following operating system platforms were used to operationally test and validate the Module to FIPS 140-2 requirements. These tested platforms provide the highest level of assurance that the module will operate correctly.

Operating System	Hardware Platform	Module File Name	CC Validation
Solaris™ 10 Release 11/06	Dell Precision 650 Workstation (Intel Xeon)	libfips.so.0.9.8	EAL4+ ALC_FLR.3 06-NOV-07 CAPP Compliant
Microsoft Windows Server 2003 SP2	Dell Optiplex GX620 (Intel Pentium 4)	libosslfips.dll	EAL4+ ALC_FLR.3 07-FEB-08 CAPP Compliant
Apple Computer Mac OS X Version 10.3.6	Power Mac G4 Dual Processor (PowerPC G4)	libfips.so.0.9.8.dylib	EAL3 13-JAN-05 CAPP Compliant
Red Hat Enterprise Linux Version 5.1	SGI Altix XE240 (Intel Xeon)	libfips.so.0.9.8	EAL4+ ALC_FLR.3 21-APR-08 CAPP Compliant
Hewlett-Packard HP-UX 11i Version 3	HP 9000 RP3440 (Intel Itanium2)	libfips.so.0.9.8	EAL4+ ALC_FLR.3 26-MAR-08 CAPP Compliant
Microsoft Corporation Windows 2000 (Server) SP3 and Q326886 Hotfix	Dell Optiplex GX400 (Intel Pentium 4)	libosslfips.dll	EAL4+ ALC_FLR.3 25-OCT-02 CAPP Compliant

FIPS 140-2 Tested Platforms

All operating systems used for module testing have been validated against the Controlled Access Protection Profile (CAPP), version 1.d, Protection Profile NoPP006 dated October 8, 1999.

In accordance with FIPS Implementation Guidance the Cimcor Cryptographic Module will remain compliant with the requirements of FIPS140-2 when operating on the following operating systems provided that the general purpose computer (GPC) uses the operating system configuration and modes specified on the referenced CC Certification Report.

OS	Configurations Evaluated in the Referenced CC Validation
Apple Computer Mac OS X Version 10.3.6 Mac OS X Server v10.3.6	Reference Section 8 of CC Certification Report http://www.commoncriteriaportal.org/files/epfiles/ST_VID4012-VR.pdf#page=23 eMac G4, iMac G3, iMac G4, iMac G5, iBook G3, iBook G4, PowerBook G3, PowerBook G4, Power Mac G3, Power Mac G4 Cube, Power Mac G4 (Single processor), Power Mac G4 Dual Processor, Power Mac G5 (Single processor), Power Mac G5 Dual Processor Power Mac G4 (Single processor), Power Mac G4 Dual Processor, Power Mac G5 (Single processor), Power Mac G5 Dual Processor, Xserve G4 (Single processor), Xserve G4 Dual Processor, Xserve G5 (Single processor), Xserve G5 Dual Processor
Solaris™ 10 Release 11/06	Reference: Section 9 of CC Certification Report http://www.commoncriteriaportal.org/files/epfiles/solaris10R1106-cert-e.pdf#page=12 Entry Level workstations and servers using an UltraSPARC II, UltraSPARC IIe, UltraSPARC IIi, UltraSPARCIII, UltraSPARCIIIi, or UltraSPARC T1 processor in a single or multiple configuration. The Netra™ 1280 and Sun Fire™ mid-frame and high-end family offering Dynamic Reconfiguration and Multiple Domaining using an UltraSPARC III Cu (copper based) or UltraSPARC IV processor. AMD based processor systems: AMD Opteron 800, 1200, and 8000 series; AMD-64 100, 200, and 2000 series; AMD dual-core 1200 and 2000 series; AMD Opteron 285; and, Intel Xeon.
Microsoft Windows Server 2003 SP2 (Includes R2, Standard, Enterprise, Datacenter, x64, and Itanium Editions; Windows XP Professional SP2 and x64 SP2; Windows XP Embedded SP2)	Reference: Section 8 of CC Certification Report http://www.commoncriteriaportal.org/files/epfiles/20080303_st_vid10184-vr.pdf#page=17 Dell Optiplex GX620; Dell PowerEdge SC1420; Dell PowerEdge SC1420; Dell PowerEdge 1800; Dell PowerEdge 28501; HP Proliant DL385; HP rx1620 Bundle Solution Server; HP xw9300 Workstation; IBM eServer 326m; IBM eServer 326m; Unisys RASCAL ES7000
Red Hat Enterprise Linux Version 5.1	Reference: Section 8 of the CC Validation Report http://www.commoncriteriaportal.org/files/epfiles/st_vid10286-vr.pdf#page=14 SGI Altix XE servers (200 and 300 series, Xeon EM64T/x86_64 based);

	SGI Altix 400 and 4000 series (Itanium2/ia64-based) consisting of a customer selected combination of the following blade types: o Compute/Memory blade o Memory-only blade o Base I/O Blade o PCI-X expansion blade o PCI-Express expansion blade
Hewlett-Packard HP UX-11i Version 3	Reference: Section III, TOE Configuration of the CC Validation Report http://www.commoncriteriaportal.org/files/epfiles/LFL-T241%20HP-UX%20CC%20CR%20v1-0.pdf#page=11 rp3410-2; rp3440-4; rp4410-4; rp4440-8; rp7420-16; rp8420-32; Superdome; BL60p; rx1620; rx2620; rx3600; rx4640; rx6600; rx7620; rx7640; rx8620
Microsoft Corporation Windows 2000 (Server) SP3 and Q326886 Hotfix (Includes Windows 2000 Server, Advanced Server, and Professional)	Reference: Section 1.1 of the CC Validation Report http://www.commoncriteriaportal.org/files/epfiles/CCEVS_VID402-VR.pdf Compaq Proliant ML570. Compaq Proliant ML330 (both 2-processor and 4-processor version). Compaq Professional Workstation AP550. Dell Optiplex GX400. Dell PE 2500. Dell PE 6450. Dell PE 2550. Dell PE 1550

CC Compatible Platforms

2.2 Ports and Interfaces

The physical ports of the Module are the same as the General Purpose Computer (GPC) on which it is executing. The logical interface is a C-language application program interface (API).

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API functions. The Status Output interface includes the return values of the API functions. In the error state data output is disabled.

Interface	***Physical Port***	***Module Interface***
Data Input	Physical Ports of a GPC	API input parameters
Data Output	Physical Ports of a GPC	API output parameters
Control Input	Physical Ports of a GPC	API function calls and parameters, other than those used by Data Input and Output interfaces
Status Output	Physical Ports of a GPC	API return codes
Power Input	Power Port of a GPC	N/A

Ports and Interfaces

3. Roles, Services, and Authentication

3.1 Roles and Services

The Module meets all FIPS 140-2 Level 3 requirements for Roles and Services, implementing both User and Crypto-Officer roles. As allowed by FIPS 140-2, the Module does not support user authentication for those roles. The operating systems hosting the Module enforce identity-based authentication to the module prior to operator access to Module functions. The operating system distinguishes between privileged administrator and non-privileged user roles. The Module has no unauthenticated services; authentication is performed by the operating system prior to module load and invocation.

The User and Crypto-Officer roles are implicitly assumed by the entity accessing services implemented by the Module. The Crypto-Officer role is implicitly entered when installing the Module or performing system administration functions on the host operating system. The Module does not implement any security relevant functions exclusive to the Crypto-Officer role, and the Crypto-Officer may access all Module services available to the User role. The allocation of functions to roles is fixed and cannot be changed by any runtime or configuration parameters.

Service	***Role***	***CSP***	***Access***
Symmetric encryption/decryption	User, Crypto-Officer	AES and TDES symmetric keys	read/write/execute volatile memory
Key wrapping for key transport	User, Crypto-Officer	RSA public/private key pairs	read/write/execute volatile memory
Key agreement primitives	User, Crypto-Officer	Diffie-Hellman keys	read/write/execute volatile memory
Digital signature	User, Crypto-Officer	RSA and DSA asymmetric keys	read/write/execute volatile memory
Symmetric key generation	User, Crypto-Officer	AES, TDES and HMAC symmetric keys	read/write/execute volatile memory
Asymmetric key generation	User, Crypto-Officer	RSA, DSA and Diffie-Hellman keys	read/write/execute volatile memory
Keyed Hash (HMAC)	User, Crypto-Officer	HMAC keys	read/write/execute volatile memory
Message digest (SHS)	User, Crypto-Officer	none	read/write/execute volatile memory
Random number generation (ANSI X9.31)	User, Crypto-Officer	RNG seed and seed key	read/write/execute volatile memory
Show status	User, Crypto-Officer	none	execute volatile memory
Module initialization	User, Crypto-Officer	none	execute volatile memory
Self-test	User, Crypto-Officer	Integrity-check HMAC key	execute volatile memory
Zeroize	User, Crypto-Officer	all symmetric and asymmetric keys, as well as parameters other than those used by Data Input and Output Interfaces	write volatile memory

Roles and Services

The Module does not store any CSPs in persistent storage, and does not read or write any persistent storage. All CSPs are stored in volatile memory only.

Note that only the private key components of public/private key pairs are CSPs. The public keys are assumed to be publicly visible.

3.2 Authentication

Authentication functions are provided by the host operating system. Authentication requires a password of at least eight characters and delay on failure as enforced by operating system configuration. Using the conservative estimate of a probability of 1/10 for a random guess for each character of such a password, the probability of guessing the password is less than one in 10,000,000 per attempt.

Given the stipulation that all user passwords must be at least 8 alphanumeric characters, the password space is in excess of 36^8 or roughly 2.8 trillion. To exceed a one in 100,000 probability of a successful guess in 60 seconds, an attacker would have to be capable of 470 thousand attempts per second, a rate which exceeds the operational capabilities of the host operating system to support.

4. Secure Operation

The tested operating systems segregate user processes into separate process spaces. Each process space is an independent virtual memory area that is logically separated from all other processes by the operating system software and hardware. The host operating system provides authentication services to prevent unauthorized access to the Module.

A complete revision history of the source code from which the Module was generated is maintained in a version control database.

Upon initialization of the Module by invoking the FIPS mode initialization API call, the module will run its power up self-tests. Specifically, the module is initialized into the FIPS mode by calling FIPS_mode_set(TRUE); a TRUE return value from this call indicates successful completion of power up self-tests and confirmation that the module is operating in the FIPS Approved mode of operation.

The self-tests can be called on demand by restarting the module (i.e., reloading the module and re-invoking the FIPS mode initialization API call). The calling application can query the current status of the FIPS mode of operation at any time.

The Module implements the following non-FIPS Approved algorithms which are allowed for use in FIPS mode: RSA encrypt/decrypt, used for key wrapping; Diffie-Hellman primitives, used for key agreement.

In the non-FIPS Approved mode of operation (prior to successful completion of the power up self-tests or subsequent to explicitly exiting the FIPS mode of operation) the same algorithms are available as in the FIPS Approved mode of operation, with the addition of algorithms and key sizes as noted in the Non-Approved Algorithms table in Section 5 below.

Invalid input via API function calls will not compromise the security of the Module.

4.1 Installation

Installation consists of copying the shared library file to the appropriate location where it can be referenced by the host operating system at application runtime.

Installation instructions:

1. Copy the shared library file to the appropriate location on the host system for protected system libraries.
2. Compute the HMAC-SHA-1 of the shared library file and compare it to the known good value supplied by the vendor to verify the integrity of that file.
3. As appropriate define or register the shared library for reference by the operating system (O/S) run-time loader. This step will vary depending on the O/S and whether the shared library is to be installed for global access by all users or only for use by a specific application. For Microsoft Windows simply placing the shared library in the same directory as the calling application suffices for use by that application. For Unix and Linux systems the LD_LIBRARY_PATH environment variable (or possibly others such as LD_PRELOAD) can be defined, or the *ldconfig* command can be used to configure the system-wide cache of shared libraries listed in the */etc/ld.so.conf* file.

4.2 Mitigation of Other Attacks

The Module does not implement security mechanisms beyond those required for FIPS 140-2 level 2 validated modules.

4.3 Physical Security

The Module is a software library and thus does not claim any physical security.

4.4 Security Rules

This section documents the security rules enforced by the cryptographic module in the general purpose computing environment to implement the security requirements of this FIPS 140-2 module.

The operating environment of the cryptographic module shall provide two distinct operator roles. These are the User role, and the Cryptographic Officer role.

The operating environment of the cryptographic module shall provide identity-based authentication: the user shall be uniquely identified by an identifier string (e.g. login name) and authenticated by verification of a eight characters minimum password.

The operating environment of the cryptographic module shall clear previous authentications on power cycle

When the operating environment of the cryptographic module has not been placed in a valid role, calling applications do not have access to any cryptographic services.

Calling applications can command the module to perform the power up self-test by invoking the FIPS mode initialization API call.

Power up self-tests do not require any action other than invoking the FIPS mode initialization API call.

Data output is inhibited during key generation, self-tests, zeroization, and error states.

For secure operation the module requires the operating environment to provide authentication event and audit mechanism access event logging.

Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module. CSPs are not stored beyond the lifetime of API function calls.

The module ensures that the seed and seed key inputs to the Approved DRNG are not equal.

There are no restrictions on which keys or CSPs are zeroized by the zeroization service.

The module does not support a maintenance interface or role.

The module does not support manual key entry.

The module is software only and does not have any external input/output devices.

The module does not enter or output plaintext CSPs except as passed by the calling application during the performance of a cryptographic operation.

The module does not output intermediate key values.

5. Cryptographic Key Management

The Module does not persistently store any Critical Security Parameters (CSPs). All CSP key input and key output operations are managed by the calling application using the Module API to transfer CSPs across the cryptographic module boundary.

CSP	Source	Usage
RSA Keys	Generated using module PRNG	Sign and verify
DSA Keys	Generated using module PRNG	Sign and verify
AES Keys	Generated using module PRNG	Encryption and decryption
TDES Keys	Generated using module PRNG	Encryption and decryption
HMAC Keys	Generated from external key and data	Message integrity
PRNG Keys	Seed, key	Key generation

CSPs

The Module supports the following FIPS Approved algorithms:

Algorithm	Certificate	Notes
AES	1121	
TDES	818	
DSA	364	1
RNG ANSI X9.31	624	
RSA (X9.31, PCKS#1.5, PSS)	530	1
SHS (SHA-1, SHA-224/256/384/512)	1044	
HMAC (HMAC-SHA-1, HMAC-SHA-224/256/384/512)	632	

Approved Algorithms

Note 1: Does not support a key size of less than 1024 bit when in the FIPS mode of operation.

The Module supports the following non-FIPS Approved algorithms:

Algorithm	Notes
RSA encrypt/decrypt	2
Diffie-Hellman primitives	3
Blowfish	4
Camellia	4
DES	4
Idea	4
RC2	4
RC4	4
RC5	4
MD2	4
MD4	4
MD5	4
Mdc2	4
Ripemd	4

Non-Approved Algorithms

Note 2: RSA (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength) allowed in FIPS mode.

Note 3: Diffie-Hellman primitives (key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength) allowed in FIPS mode.

Note 4: Available in non-FIPS mode only.

5.1 Key Zeroization

Keys residing in internally allocated data structures can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access.

Only the process that creates or imports keys can use or export them. No persistent storage of key data is performed by the Module. All API functions are executed by the invoking process in a non-overlapping sequence such that no two API functions will execute concurrently. The Module zeroizes dynamically allocated volatile memory when de-allocating that memory.

The Module provides a zeroization function for use by calling applications. Rebooting of the system will zeroize any keys present in volatile RAM.

5.2 Self-Tests

The Module performs both power up self-tests at module initialization and conditional tests during operation. Input, output, and cryptographic functions cannot be performed while the Module is in a self-test or error state as the module is single threaded and will not return to the calling application until the power up self-tests are complete. If the power up self-tests fail subsequent calls to the module will fail and thus no further cryptographic operations are possible.

Power Up Self-Tests

Algorithm	Test
AES	KAT
Triple-DES	KAT
DSA signatures	pairwise consistency test, sign/verify
RSA signatures	KAT
ANSI X9.31 PRNG	KAT
HMAC-SHA-1	KAT
HMAC-SHA-224	KAT
HMAC-SHA-256	KAT
HMAC-SHA-384	KAT
HMAC-SHA-512	KAT
SHA-1	KAT[1]
SHA-224	KAT[1]
SHA-256	KAT[1]
SHA-384	KAT[1]
SHA-512	KAT[1]
Module integrity check	HMAC-SHA-1

Power Up Self-Tests

Conditional Self-Tests

Algorithm	Test
DSA key pair generation	pairwise consistency
RSA key pair generation	pairwise consistency
PRNG	continuous RNG test

Conditional Self-Tests

A single API call is required to initialize the Module for operation in the FIPS 140-2 Approved mode. When so initialized the Module performs all security functions and cryptographic algorithms in FIPS Approved mode.

The initialization function verifies the integrity of the runtime executable using a HMAC-SHA-1 digest computed at build time. If this computed HMAC-SHA-1 digest matches the stored known digest then the power up self-test, consisting of the algorithm specific Pairwise Consistency and Known Answer tests, is performed. If any component of the power up self-test fails subsequent invocation of any cryptographic function calls is disabled. Any such failure is a hard error that can only be recovered by reloading the Module.

[1] Tested as part of the HMAC known answer tests.