FIPS 140-2 Non-Proprietary Security Policy

Cocoon Data Secure Objects C++ Cryptographic Module Version 1.8

Document Version 1.7

11 March 2013

Prepared For:

Cocoon Data

Level 4

152-156 Clarence St

Sydney – NSW – 2000

www.cocoondata.com

Prepared By:

Apex Assurance Group, LLC

530 Lytton Avenue

Ste. 200

Palo Alto, CA 94301

www.apexassurance.com

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the Secure Objects C++ Cryptographic Module Version 1.8.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 About FIPS 140

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Communications Security Establishment of Canada (CSEC) Cryptographic Module Validation Program (CMVP) runs the FIPS 140 program. The CMVP accredits independent testing labs to perform FIPS 140 testing; the CMVP also validates test reports for modules meeting FIPS 140 validation. *Validated* is the term given to a product that is documented and tested against the FIPS 140 criteria.

More information is available on the CMVP website at
http://csrc.nist.gov/groups/STM/cmvp/index.html.

1.2 About this Document

This non-proprietary Cryptographic Module Security Policy for the Secure Objects C++ Cryptographic Module Version 1.8 from Cocoon Data provides an overview of the product and a high-level description of how it meets the security requirements of FIPS 140-2. This document contains details on the module's cryptographic keys and critical security parameters. This Security Policy concludes with instructions and guidance on running the module in a FIPS 140-2 mode of operation.

The Cocoon Data Secure Objects C++ Cryptographic Module Version 1.8 may also be referred to as the "module" in this document.

1.3 External Resources

The Cocoon Data website (http://www.cocoondata.com) contains information on Cocoon Data products. The Cryptographic Module Validation Program website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2013.htm) contains links to the FIPS 140-2 certificate and Cocoon Data contact information.

1.4 Notices

This document may be freely reproduced and distributed in its entirety without modification.

1.5 Acronyms

The following table defines acronyms found in this document:

Acronym	Term
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
GUI	Graphical User Interface
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
MD	Message Digest
NIST	National Institute of Standards and Technology
OS	Operating System
PKCS	Public-Key Cryptography Standards
PRNG	Pseudo Random Number Generator
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security
USB	Universal Serial Bus

Table 1 – Acronyms and Terms

2 Cocoon Data Secure Objects C++ Cryptographic Module Version 1.8

2.1 Solution Overview

The Cocoon Data Secure Objects C++ Cryptographic Module Version 1.8 has been implemented as part of the Cocoon Data Secure Objects solution, an encryption-based access control system for protecting the confidentiality and integrity of electronic files. Secure Objects controls and monitors the exchange of digital files based on recipient identity, to protect against the deliberate or unintentional release of sensitive user data.

2.2 Cryptographic Module Specification

The module, the Secure Objects C++ Cryptographic Module Version 1.8, is a software shared library that provides cryptographic services required by the Cocoon Data Secure Objects solution. The Module's logical cryptographic boundary is the shared library files and their integrity check HMAC files, which are as follows:

- libcrypto-macOS.dylib
- libcrypto-macOS.dylib.hash (cdfa5dd04b37ec9b72b90a9d0aff4b5a262755d2)

- libcrypto32-ubuntu.so
- libcrypto32-ubuntu.so.hash (84ba4d785833d29281ce3ec7dd8b2ea2ab85b7f2)

- libcrypto64-ubuntu.so
- libcrypto64-ubuntu.so.hash (f6b60bd790c48cc36243da24512d303598eadc3e)

- libcrypto32-redhat.so
- libcrypto32-redhat.so.hash (153ffe5af859fa182f90f8c72710c3a0c9316463)

- libcrypto64-redhat.so
- libcrypto64-redhat.so.hash (c31e2f2ab982364b62aab779ceddbd26f10c3288)

- WindowsFIPSx32vs10.dll
- WindowsFIPSx32vs10.dll.hash (ed8e94aff4e7f3529c193d8bd263566aebf07d23)

- WindowsFIPSx64vs10.dll
- WindowsFIPSx64vs10.dll.hash (a830035554655da14deff9e36dc096be5d688924)

- WindowsFIPSx32vs12.dll
- WindowsFIPSx32vs12.dll.hash (748007973df84b6144a1838f6c89bd2a5ca009ab)

- WindowsFIPSx64vs12.dll
- WindowsFIPSx64vs12.dll.hash (0a1739fa8e495e61ffcae65ffc0887c204e63dbc)

The module is a multi-chip standalone embodiment installed on a General Purpose Computer. All operations of the module occur via calls from the Cocoon Data applications and their respective internal daemons/processes. As such there are no untrusted services calling the services of the module, as APIs are not exposed.

2.2.1 Validation Level Detail

The following table lists the level of validation for each area in FIPS 140-2:

FIPS 140-2 Section Title	Validation 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
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A
Overall Level	**1**

Table 2 – Validation Level by DTR Section

2.2.2 Approved Cryptographic Algorithms

The module's cryptographic algorithm implementations have received the following certificate numbers from the Cryptographic Algorithm Validation Program:

FIPS 140-2 Section Title	CAVP Certificate
AES	#2192
TDES	#1385
SHS	#1900
HMAC	#1344
HMAC_DRBG: SP800-90 (hash based)	#257

Table 3 – FIPS-Approved Algorithm Certificates

2.2.3 Non-Approved Cryptographic Algorithms

The module does not support any non-approved algorithms.

2.3 Module Interfaces

The figure below shows the module's physical and logical block diagram:

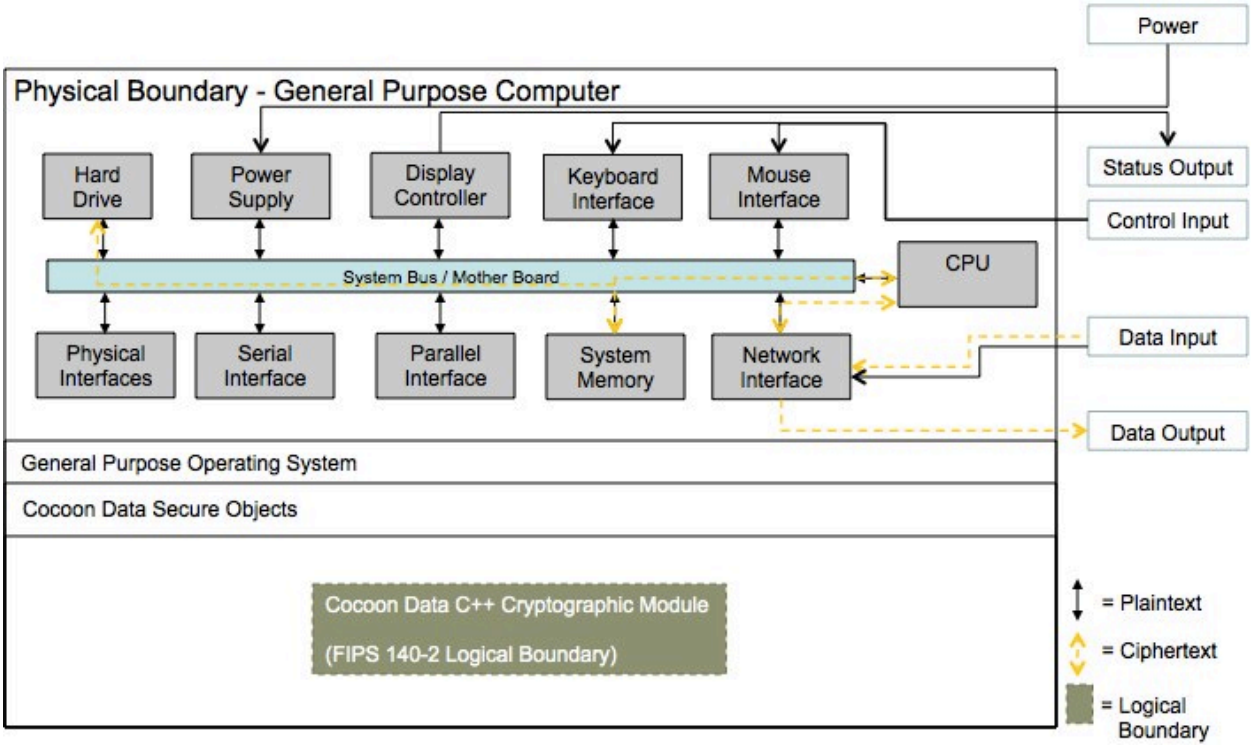

Figure 1 – Module Boundary and Interfaces Diagram

The interfaces (ports) for the physical boundary include the computer keyboard port, CDROM drive, floppy disk, mouse, network port, parallel port, USB ports, monitor port and power plug. The module has logical interfaces provided through the Application Programming Interface (API) that a calling daemon can operate. The logical interfaces expose services that applications directly call, and the API provides functions that may be called by a referencing application (see Section 2.4 – Roles, Services, and Authentication for the list of available functions). The module distinguishes between logical interfaces by logically separating the information according to the defined API.

The API provided by the module is mapped onto the FIPS 140- 2 logical interfaces: data input, data output, control input, and status output. Each of the FIPS 140- 2 logical interfaces relates to the module's callable interface, as follows:

FIPS 140-2 Interface	Logical Interface	Module Physical Interface
Data Input	Input parameters of API function calls	Network Interface
Data Output	Output parameters of API function calls	Network Interface

FIPS 140-2 Interface	Logical Interface	Module Physical Interface
Control Input	API function calls	Keyboard Interface, Mouse Interface
Status Output	For FIPS mode, function calls returning status information and return codes provided by API function calls.	Display Controller
Power	None	Power Supply

Table 4 – Logical Interface / Physical Interface Mapping

As shown in Figure 1 – Module Boundary and Interfaces Diagram and Table 6 – Module Services and Descriptions, the output data path is provided by the data interfaces and is logically disconnected from key management processes. No key information will be output through the data output interface when the module zeroizes keys.

2.4 Roles, Services, and Authentication

The module supports a Crypto Officer and a User role. The module does not support a Maintenance role. The supported role definitions are as follows:

Role	Services
User	Encryption, Decryption (symmetric), Random Numbers
Crypto Officer	Installation and configuration of FIPS 140-2 validated mode

Table 5 – Role Descriptions

The User and Crypto-Officer roles are implicitly assumed by the entity accessing services implemented by the Module.

2.4.1 Operator Services and Descriptions

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 Design documentation. The following table shows the services available to the various roles and the access to cryptographic keys and CSPs resulting from services:

Service	Roles	KEY / CSP	Permission
Symmetric encryption/decryption	User	AES Key, Triple-DES Key	User: write/execute
Message digest (SHS)	User	none	User: na
Show status	User	none	User: na
Module initialization	CO	none	CO: na

Service	Roles	KEY / CSP	Permission
Self-Test	CO	All CSPs	CO: execute
Zeroize	User	All CSPs	User: write
HMAC_Drbg Generate random number.	User	HMAC_DRBG V HMAC_DRBG C	User: write/execute
HMAC	USER	HMAC Key for Integrity Check	User: execute

Table 6 – Module Services and Descriptions

2.4.2 Operator Authentication

As required by FIPS 140-2, there are two roles (a Crypto Officer role and User role) in the module that operators may assume. As allowed by Level 1, the module does not support authentication to access services. As such, there are no applicable authentication policies. Access control policies are implicitly defined by the services available to the roles as specified in Table 6 – Module Services and Descriptions.

2.5 Physical Security

This section of requirements does not apply to this module. The module is a software-only module and does not implement any physical security mechanisms.

2.6 Operational Environment

The module operates on a general purpose computer (GPC) running a general purpose operating system (GPOS). For FIPS purposes, the module is running on this operating system in single user mode and does not require any additional configuration to meet the FIPS requirements.

The module was tested on the following platforms:

- Microsoft Windows 7 32-bit with MSVC2010 redistributable running on Dell Vostro 1520 (single user mode)
- Microsoft Windows XP 32-bit with SP3; MSVC2010 redistributable running on Dell Vostro 1520 (single user mode)
- Microsoft Windows 7 64-bit with MSVC2010 redistributable running on Dell Vostro 3500 (single user mode)

- Microsoft Windows 7 32-bit with MSVC2012 redistributable running on Dell Vostro 1520 (single user mode)
- Microsoft Windows XP 32-bit with SP3; MSVC2012 redistributable running on Dell Vostro 1520 (single user mode)
- Microsoft Windows 7 64-bit with MSVC2012 redistributable running on Dell Vostro 3500 (single user mode)
- Ubuntu 12.04 LTS 64-bit running on Dell PowerEdge 1950
- Ubuntu 12.04 LTS 64-bit on VMWare Fusion 4.1.3 on OSX running on a MacBook Pro Intel core i7
- Ubuntu 12.04 LTS 32-bit running on Dell PowerEdge 1950
- Ubuntu 12.04 LTS 32-bit on VMWare Fusion 4.1.3 on OSX running on a MacBook Pro Intel Core i7
- Redhat Enterprise Linux Server 6.3 64-bit running on Dell PowerEdge 1950
- Redhat Enterprise Linux Server 6.3 64-bit on VMWare Fusion 4.1.3 on OSX running on a MacBook Pro Intel Core i7
- Redhat Enterprise Linux Server 6.3 32-bit running on Dell PowerEdge 1950
- Redhat Enterprise Linux Server 6.3 32-bit on VMWare Fusion 4.1.3 on OSX running on a MacBook Pro Intel Core i7
- Mac OSX 10.8 running on MacBook Pro Intel Core i7

The GPC(s) used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B. FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the GPOS running in single user mode, assuming that the requirements outlined in NIST IG G.5 are met.

2.7 Cryptographic Key Management

The table below provides a complete list of Critical Security Parameters used within the module:

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
AES Key	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD

Keys and CSPs	Storage Locations	Storage Method	Input Method	Output Method	Zeroization	Access
Triple-DES Key	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC Key	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC_DRBG V	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC_DRBG C	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMACs_DRBG Entropy Input String	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC_DRBG Seed Value	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC Key	RAM	Plaintext	API call parameter	None	`ClearSensitiveData()` power cycle	U: RWD
HMAC Key for Integrity Check	RAM	Plaintext	API call parameter	None	`uninstall`	U: RWD

R = Read W = Write D = Delete

Table 7 – Module Keys/CSPs

The application that uses the module is responsible for appropriate destruction and zeroization of the key material. The library provides functions for key allocation and destruction which overwrite the memory that is occupied by the key information with zeros before it is deallocated.

2.7.1 Key/Critical Security Parameter (CSP) Authorized Access and Use by Role and Service/Function

The module does not provide any key generation services. Key and CSP data can be provided to the module but there is no interface for later retrieving (regardless of Role).

2.7.2 Key/CSP Storage

Keys are provided to the Module by the calling process, and are destroyed when released by the appropriate API function calls. The Module does not perform persistent storage of keys.

2.7.3 Key/CSP Zeroization

The memory occupied by keys is cleared by a destruction function (via the `ClearSensitiveData()` call) that overwrites the memory occupied by keys with zeros, which sufficiently protects the CSPs from compromise.

2.7.4 Key Generation

The module does not support key generation

2.8 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. All of these tests are listed and described in this section. In the event of a self-test error, the module will log the error and will halt. The module must be initialized into memory and pass the self-tests to resume function.

No operator intervention is required during the running of the self-tests. Self-tests can be performed on demand by functions provided as described in the Cryptography Design and API documents.

The following sections discuss the module's self-tests in more detail.

2.8.1 Power-On Self-Tests

Power-on self-tests are executed automatically when the module is loaded into memory via the `FIPS_instance()` method. The `doModuleIntegrityCheck()` function verifies the integrity of the runtime executable using a HMAC SHA-1 digest computed at build time. If the digests match, the power-up self-tests are then performed. If the power-up self-test is successful, `FipsFactory::Instance()` sets an enumerated value `testStatus` to PASSED and the Module is in FIPS mode.

TYPE	DETAIL
Software Integrity Check	HMAC SHA-1
Known Answer Tests	• AES encrypt/decrypt • TDES encrypt/decrypt • HMAC SHA-1 • HMAC SHA-224 • HMAC SHA-256 • HMAC SHA-384 • HMAC SHA-512 • SHA-1 • SHA-224 • SHA-256 • SHA-384 • SHA-512 • AES encrypt/decrypt Monte Carlo • TDES encrypt/decrypt Monte Carlo • HMAC_DRBG tests

Table 8 – Power-On Self-Tests

Input, output, and cryptographic functions cannot be performed while the Module is in a self-test or error state because 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 also fail - thus no further cryptographic operations are possible without resetting the module.

2.8.2 Conditional Self-Tests

The module implements a continuous RNG test on the HMAC_DRBG implementation.

2.8.3 Critical Functions Tests

The module does not perform critical functions tests above and beyond power-on self-tests and conditional self-tests, as there are no other functions that, upon failure, could lead to the disclosure of CSPs.

2.9 Mitigation of Other Attacks

The Module does not contain additional security mechanisms beyond the requirements for FIPS 140-2 Level 1 cryptographic modules.

3 Guidance and Secure Operation

This section describes how to configure and initialize the module for FIPS-Approved mode of operation. When configured and initialized per this Security Policy, the module will only operate in the FIPS Approved mode of operation.

3.1 Crypto Officer Guidance

3.1.1 Enabling FIPS Module within the Secure Objects Application

The module is included with the Secure Objects suite including Secure Envelopes and Secure Documents and is not available for direct download. The FIPS Mode setting is configured in development when calling Secure Objects C++ Cryptographic Module. The Secure Objects application is configured to use the module as follows:

- The Cocoon Data development team is responsible for ensuring the source files that comprise the Secure Objects C++ Cryptographic Module Version 1.8 are built into the Secure Objects solution.
- The module ships in FIPS mode by default, and there is no non-FIPS mode.

3.1.2 Additional Rules of Operation

1. The writable memory areas of the Module (data and stack segments) are accessible only by the Secure Objects application so that the operating system is in "single user" mode, i.e. only the Secure Objects application has access to that instance of the Module.
2. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the Module.
3. The end user of the operating system is also responsible for zeroizing CSPs via wipe/secure delete procedures. These procedures are commands via operating system or third party applications to wipe disk space, and they augment the zeroization functions implemented within the module.

3.2 User Guidance

3.2.1 General Guidance

The module is not distributed as a standalone library and is only used in conjunction with the Cocoon Data solution. As such, there is no direct User Guidance.