Comtech Mobile Datacom Corporation Cryptographic Library `libcmscrypto`

(Version 1.0)

FIPS 140-2 Non-Proprietary Security Policy

Level 1 Validation

Document Version 1.4

Revision History

Revision	Revision Date	Description of the Changes
1.0	April 8, 2009	Initial document, all sections affected
1.1	May 1, 2009	Edits and format changes
1.2	May 21, 2009	Added block diagram of the module and a list of API functions in table 4
1.3	June 3, 2009	Added AES CFB128 encryption/decryption. TDES CBC CTS listed as non FIPS approved mode
1.4	July 6, 2009	Various updates based on lab comments

Table of Contents

Figures

Tables

1 INTRODUCTION

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Comtech Mobile Datacom Corporation Cryptographic Library (libcmscrypto), which is predominantly used for Comtech Mobile Messaging System (CMS) applications from Comtech Mobile Datacom Corporation. This Security Policy describes how the libcmscrypto Cryptographic Library meets the security requirements of Federal Information Processing Standards (FIPS) 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – Security Requirements for Cryptographic Modules) details the U.S. 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) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/groups/STM/cmvp/

The libcmscrypto Cryptographic Library is referred to in this document as, the Cryptographic Library, 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 Comtech website (http://www.comtechmobile.com) contains information on the full line of products from Comtech.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/) contains contact information for answers to 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 Comtech Mobile Datacom Corporation. With the exception of this Non-Proprietary

Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Comtech and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Comtech.

2 LIBCMSCRYPTO CRYPTOGRAPHIC LIBRARY

2.1 Overview

The Comtech Mobile Datacom Corporation Cryptographic Library (libcmscrypto) is a generic module that contains Application Programming Interface (API) functions used to provide support for cryptography for the CMS network applications. All encryption and decryption in the CMS network applications is performed through the API of the cryptographic module.

2.2 Module Specification

The cryptographic module is the "libcmscrypto.so" library version 1.0, generated from a set of C++ language source files. The module was tested and validated on Red Hat Enterprise Linux v5.0 running on an Intel Celeron processor. Comtech affirms that the module can be run unaltered on any version of Red Hat Enterprise Linux v4.6 or higher running on an Intel or AMD processor and maintain compliance to FIPS 140-2. The module provides an API for invocation of FIPS approved and non-approved cryptographic functions from the calling applications. The module 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 module is classified as multi-chip standalone cryptographic module implemented completely in software for FIPS-140-2 purposes. The physical boundary of the libcmscrypto Cryptographic Library for CMS is defined by the enclosure of the computer system on which it is executing. The logical cryptographic boundary of the module is the libcmscrypto.so file itself.

The diagram below identifies all software and hardware components

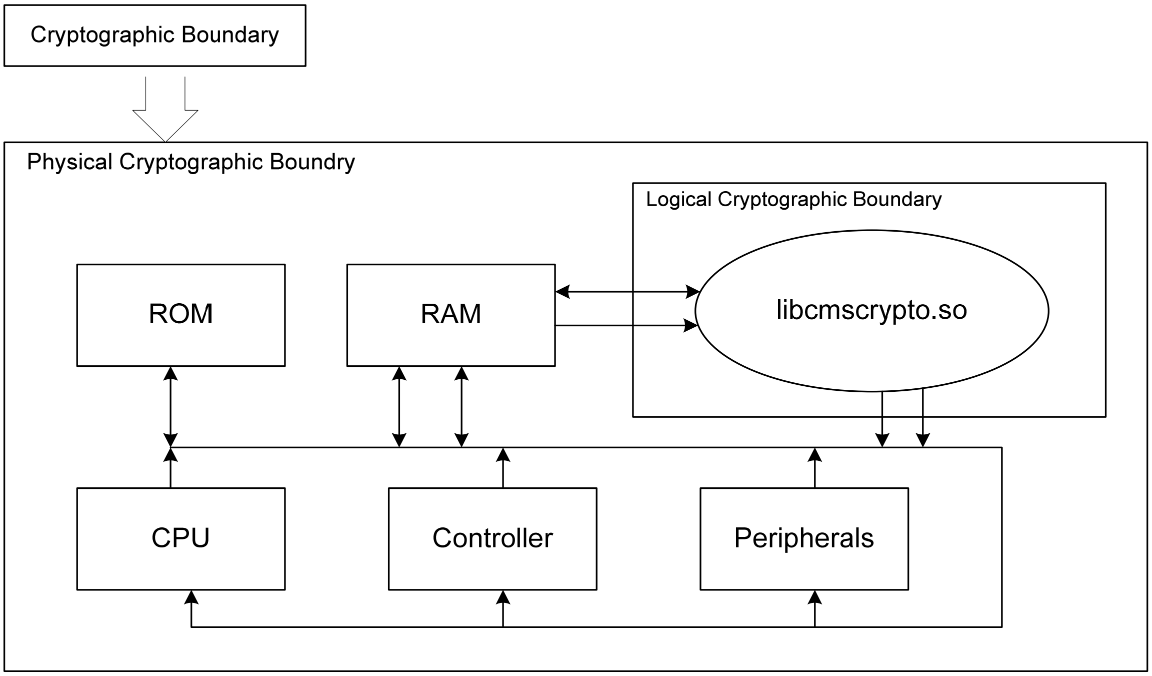

Figure 1: Block diagram of hardware and software components

The cryptographic module boundary consists of the following components[1]:

- DES symmetric block cipher that uses Data Encryption Algorithm (DES)
- Triple DES (TDES) symmetric block cipher module that uses Triple Data Encryption Algorithm (TDEA)
- AES symmetric block cipher module that uses Advanced Encryption Standard (AES)
- SHA1 module that uses Secure Hash Algorithm (SHA1)
- HMAC SHA1 module that uses Keyed-Hashing Message Authentication Code (HMAC)

The module 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

[1] See section 2.6 for a listing of which algorithms are allowed in a FIPS mode of operation

Section	Section Title	Level
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	Electromagnetic Interference (EMI)/ Electromagnetic Compatibility (EMC)	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.3 Roles and Services

There are two main roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto-Officer role and User role. They are the operators on the Linux Operating System on which the module is validated. The operator implicitly assumes either role based upon the services being performed. The Crypto-Officer is responsible for installation, initialization of the module and configuring it to run in a FIPS approved mode. After the switch to the FIPS approved mode of operation both the Crypto-Officer and User will be able to utilize the functionality of the module. The various services offered by the module are described below.

Table 2: Services Authorized for Roles

Role	Authorized Services
User role	All services except installation and initialization
Crypto Officer role	All services including installation and initialization

Table 3: Access Rights within Services

Service	Role	Key/CSP	API functions	CSP Access
Symmetric encryption/decryption	User, Crypto-Officer	AES, TDES or DES key	aes_encrypt aes_decrypt generate_aes_subkeys aes_encrypt_CFB128 aes_decrypt_CFB128	read/write/execute

			aes_encrypt_CBC_CT aes_decrypt_CBC_CT DES_block c_DES_block generate_c_subkeys c_triple_DES_block tdes_encrypt_CBC_CT tdes_decrypt_CBC_CT	
Keyed Hash MAC	User, Crypto-Officer	HMAC key	hmac	read/write/execute
Message digest	User, Crypto-Officer	none	SHA1_Init SHA1Update SHA1_Final	N/A
Module initialization and installation	Crypto-Officer	none	get_versions get_fips_mode fips_mode_set	N/A
Show status	Crypto-Officer	none	get_fips_mode fips_mode_set errormsg	N/A
Self-test	User, Crypto-Officer	Integrity Key	fips_mode_set fips_selftests fips_integrity_check fips_selftest_aes kat_selftest_tdes fips_selftest_sha1 fips_selftest_hmac	execute
Zeroize	User, Crypto-Officer	AES, TDES or DES key; HMAC key	zeroize	N/A

2.3.1 Authentication Mechanisms

The module doesn’t support authentication mechanisms.

2.4 Physical Security

The module is implemented completely in software and physical security provided solely by the host platform.

2.5 Operational Environment

The module was validated running on Red Hat Enterprise Linux v5.0; however, compliance can be maintained by running the unmodified module binary on any version of Red Hat Enterprise Linux v4.6 or higher. The Object Module functions are completely within the process space of the application process which loads it. It 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, so while cryptographic processing is in use, keys and CSPs are protected by process separation.

2.6 Cryptographic Key Management

The cryptographic module implements the following FIPS-approved algorithms:

- AES (Certificate #1124)
- SHA-1 Byte oriented (Certificate #1047)
- HMAC SHA-1 (Certificate #635)

The cryptographic module implements the following non-FIPS-approved algorithms:

- Data Encryption Standard (DES)
- Triple DES CBC with Ciphertext Stealing (CTS); 1 and 2 keying option;

The module supports the following critical security parameters:

Table 4: List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
AES key	Symmetric AES (128, 192, 256)-bit key	Generated externally; input in plaintext	Never output from module	Stored in RAM in plaintext	Erasing from RAM	Used by external application to encrypt and decrypt data
TDES key (used as non-FIPS approved mode)	Symmetric TDES 192-bit key	Generated externally; input in plaintext	Never output from module	Stored in RAM in plaintext	Erasing from RAM	Used by external application to encrypt and decrypt data
DES key (used as non-FIPS approved mode)	Symmetric DES 56-bit key	Generated externally; input in plaintext	Never output from module	Stored in RAM in plaintext	Erasing from RAM	Used by external application to encrypt and decrypt data

Key	**Key Type**	**Generation / Input**	**Output**	**Storage**	**Zeroization**	**Use**
HMAC key	Variable length (min. 10-bytes required in FIPS mode)	Generated externally; input in plaintext	Never output from module	Stored in RAM in plaintext	Erasing from RAM	Used by external application to perform data integrity
Integrity Key	20-byte HMAC Key	Hardcoded in module binary	Never output from module	Non-volatile storage of GPC in plaintext	Removal from non-volatile storage and formatting	Used to perform the software integrity test

2.6.1 Key Generation

The module does not generate any cryptographic keys internally.

2.6.2 Key Storage

The module does not provide any persistent key storage. All keys are entered the module via API calls and stored in RAM in plaintext during API function execution.

2.6.3 Key Entry and Output

All keys and CSPs that are entered into the module are electronically entered in plaintext via parameters of API functions. All keys never output from the module.

2.6.4 Key Zeroization

A user can perform key zeroization of AES, TDES or DES keys by calling an API function that erases keys from the RAM.

2.7 Self-Tests

The Module performs the following power-up self-tests to ensure correct operation:

- Software integrity check using HMAC SHA1
- Known Answer Tests (KATs)
  - AES KAT
  - SHA-1 KAT
  - HMAC SHA1 KAT

Power-up tests are performed automatically when the module is loaded and invoked in FIPS mode of operation by calling the fips_mode_set() function. The integrity of the module is verified by calculating the SHA1 HMAC digest of the libcmscrypto.so file and comparing it to the digest that was calculated and placed into libcmscrypto.hmac file during the last module build-up and installation. FIPS mode cryptographic functionality will be available only after successful execution of all power-up tests.

The failure of any power-up self-tests causes the module to enter Error state (see Finite State Model) and all cryptographic operations are disabled until the module is initialized again with successful execution of self-tests.

2.8 Design Assurance

The source code is primarily written in C++. All source code and documentation is stored in CVS source code control system.

Additionally, Borland StarTeam 2008 Release 2 is used to provide configuration management for the Libcmscrypto Cryptographic library’s FIPS documentation. This software provides access control, versioning, and logging.

2.9 Mitigation of Other Attacks

The module does not mitigate other attacks.

3 SECURE OPERATION

The 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 Crypto-Officer Guidance

The Crypto-Officer is required to install and initialize the module to run in a FIPS mode of operation. The FIPS mode initialization is performed when the application invokes the fips_mode_set() call that will run power-up self-tests. Successful completion of the power-up self-tests ensures that module is operating in FIPS mode of operation.

3.1.1 Initial Setup

The cryptographic module is the libcmscrypto.so object library, version 1.0 that is installed on Linux operating system Red Hat Enterprise Linux v4.6 or higher[2]. The operating system segregates 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 module functions entirely within the process space of the CMS application process that invokes it and satisfies the FIPS 140-2 requirement for a single user mode of operation. The module is installed in the standard library path on the destination platform by copying it to the appropriate location. The installation of the module (libcmsrypto.so) also includes a HMAC-SHA1

[2] FIPS 140-2 validation completed on Red Hat Enterprise Linux v5.0

digest of the libcmscrypto.so module that is located in the libcmscrypto.hmac file. The digest is used to verify module integrity during startup.

3.2 User Guidance

The User does not have the ability to initialize and install the module. The User can use the API to encrypt and decrypt data, calculate hash digests, or calculate HMAC values. The user should only use those algorithms approved for use in the FIPS mode of operation (AES, HMAC and SHA-1).

4 ACRONYMS

The following table spells out the acronyms used in this document.

Table 5: Acronyms used in this document

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CMVP	Cryptographic Module Validation Program
CMS	Comtech Mobile Messaging System
CSP	Critical Security Parameter
DES	Data Encryption Standard
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
OS	Operating System
CPU	Central Processing Unit

Acronym	**Definition**
RAM	Random-Access Memory
SHA	Secure Hash Algorithm
TDEA	Triple Data Encryption Algorithm
TDES	Triple DES