CoCo Cryptographic Module 2.0

FIPS 140-2 Security Policy

Version No.: 2.13
Date: October 8, 2013

Prepared by:
CoCo Communications Corp.
www.cococorp.com
800 5th Avenue, Suite 3700
Seattle, WA 98104

Table of Contents

List of Figures

List of Tables

1 Introduction

This document is a non-proprietary FIPS 140-2 Security Policy for the CoCo Cryptographic Module 2.0 (the Module). It contains a specification of the rules under which the Module must operate and describes how the Module meets the requirements as specified in Federal Information Processing Standards Publication 140-2 (FIPS PUB 140-2) for a Security Level 1, multi-chip, standalone software module.

1.1 Purpose of the Security Policy

There are three major reasons why a security policy is requested:

- It is required for FIPS 140-2 validation.
- It allows individuals and organizations to determine whether the cryptographic module, as implemented, satisfies the stated security policy.
- It describes the capabilities, protections, and access rights provided by the cryptographic module that will allow individuals and organizations to determine whether it meets their security requirements.

1.2 Target Audience

This document will be one of many that are submitted as a package for FIPS validation; it is intended for the following people:

- Developers working on the release.
- The FIPS 140-2 testing lab.
- Cryptographic Module Validation Program (CMVP).
- Consumers.

2 Cryptographic Module Specification

This document is the non-proprietary security policy for the CoCo Cryptographic Module 2.0, and was prepared as part of the requirements process that will ensure its conformance with Federal Information Processing Standard (FIPS) 140-2, Level 1. The following section describes the Module and how it complies with the FIPS 140-2 standard in each of the required areas.

2.1 Module Description

Table 1: Security Levels provides an overview of the security level required for each validation section.

Security Component	Security 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
EMI/EMC	1
Self Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1: Security Levels

The Module has been tested on the platforms shown in Table 2: Tested Platforms

.

Module/Implementation	Processor	OS and Version	Test Platform
CoCo Crypto Module 2.0	AMD Geode	Linux 2.6 32-bit (single-user mode)	oMG 2000
CoCo Crypto Module 2.0	Intel x86	Vyatta 6.4 32-bit (single-user mode)	Dell PowerEdge R210

Table 2: Tested Platforms

2.2 Description of Approved Mode

The Module supports only the Approved mode and provides support for the following approved functions:

- AES (CCM, ECB , CBC, CTR, GCM)
- TDES(ECB, CBC)
- HMAC (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- SHS (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- SHA-1 (for integrity check only, Cert.#1982, Cert.#1983)
- HMAC-SHA-1 (for integrity check only, Cert.#1413, Cert.#1414)

2.3 Cryptographic Module Boundary

The logical boundary of the module is the binary code of the CoCo Cryptographic Module 2.0. Its distribution package file is :

crypto-loader_2.0.831_i386.deb for Vyatta 6.4

and

crypto-loader-2.0-831coco.i586.rpm for Linux 2.6

Figure 1 shows the logical boundary of the module's software components.

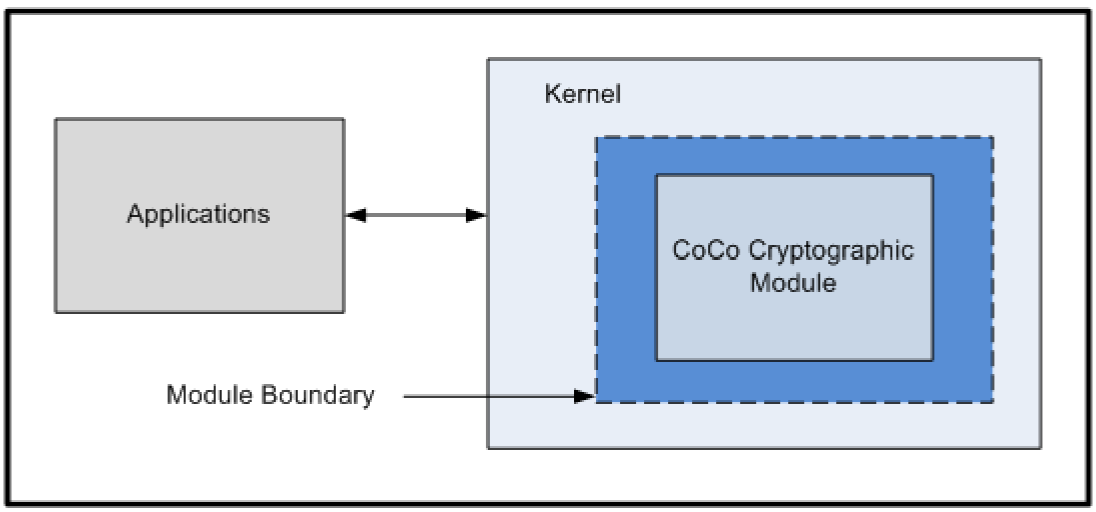

Figure 1: Software Block Diagram

The physical boundary of the module is the enclosure of the test platform on which the software module executes. Figure 2 shows the physical boundary of the module and hardware components of the platforms on which the module executes.

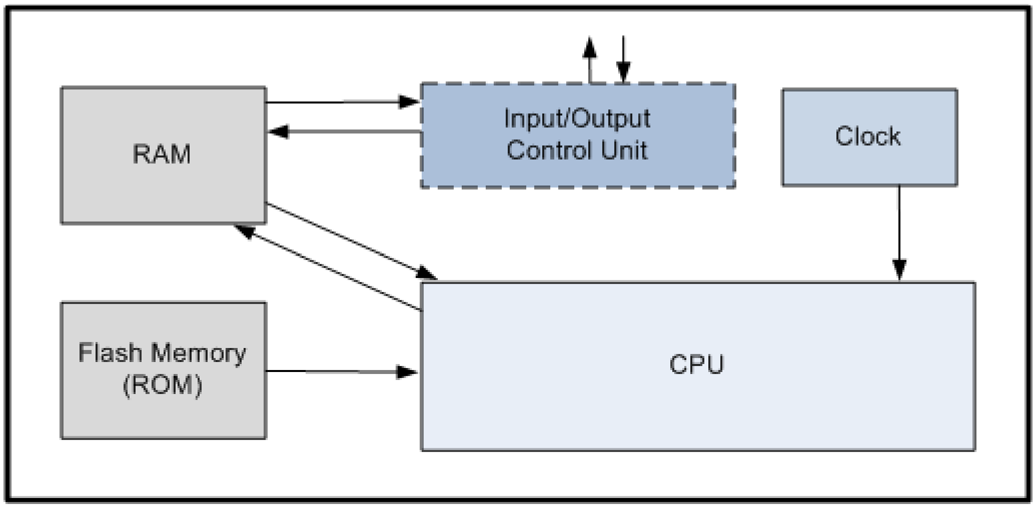

Figure 2: Hardware Block Diagram

3 Cryptographic Module Ports and Interfaces

Table 3: Ports and Interfaces shows which FIPS interfaces and ports the Module utilizes.

FIPS Interface	Ports
Data Input	API input parameters
Data Output	API output parameters
Control Input	API function calls, HMAC-SHA-1 value in the binary code
Status Output	API return codes, kernel log files, kernel process files
Power Input	Physical power connector

Table 3: Ports and Interfaces

4 Roles, Services, and Authentication

4.1 Roles

The User and Crypto Officer roles are implicitly assumed by the entity that is accessing services implemented by the Module, so no further authentication is required. The services associated with each role are explained in the next section.

4.2 Services

Service	Roles		CSP	Modes	FIPS Approved (Cert #)	Standard	API Functions
	User	CO					
Service Provided via Symmetric Algorithms							
AES Encryption Input: *plaintext, IV, key* Output: *ciphertext* Decryption Input : *ciphertext, IV, key* Output: *plaintext*	✓		128-, 192-, 256-bit keys	ECB, CBC, CTR	(Cert # 2299) -AMD Geode (Cert # 2300) -Intel x86	FIPS 197	All API functions with prefix fips_crypto_cipher_, fips_crypto_ablkcipher_ and fips_crypto_blkcipher_ ablkcipher_request_set_tfm ablkcipher_request_free ablkcipher_request_set_callback ablkcipher_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher
TDES Encryption Input: *plaintext, IV, key* Output: *ciphertext* Decryption Input :	✓		K1, K2, K3 independent	ECB, CBC	(Cert # 1446) -AMD Geode (Cert # 1447) -Intel x86	SP 800-67	All API functions with the prefix of fips_cryp-to_cipher_, fips_cryp-to_ablkcipher_ and fips_crypto_blkcipher_ cryp-to_free_ablkcipher crypto_has_ablkcipher ablkciph-er_request_set_tfm ablkciph-er_request_free ablkciph-er_request_set_callback ablkciph-

ciphertext, IV, key Output: *plaintext*							er_request_set_crypt crypto_free_blkcipher crypto_has_blkcipher
GCM <u>Encryption</u> Input: *plaintext, IV, key, AAD* Output: *Ciphertext* <u>Decryption</u> Input : *ciphertext, IV, key, AAD* Output: *plaintext*	✓		128-, 192-, 256-bit keys 96-bit IV supported Max IV length: 1024	Tag length supports 32, 63, 96, 104, 112, 120, and 128	(Cert # 2299) -AMD Geode (Cert # 2300) -Intel x86	SP 800-38D	All API functions with prefix fips_crypto_gcm
Hash Function Services							
SHA-1 SHA-224 SHA-256 SHA-384 SHA-512 Input: *message* Output: *message digest*	✓			N/A	(Cert # 1980) -AMD Geode (Cert # 1981) -Intel x86	FIPS 180-4	All API functions with prefix fips_crypto_hash fips_crypto_free_hash
Message Authentication Code (MAC) Services							
HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-225 HMAC-SHA-384 HMAC-SHA-512 Input: *HMAC key, message*	✓				(Cert # 1411) -AMD Geode (Cert # 1412) -Intel x86	FIPS 198	API functions with prefix fips_crypto_shash ,hmac_ fips_crypto_free_hash

Output: *HMAC value of the message*							
CCM <u>Encryption</u> Input: *plaintext, IV, key, AAD* Output: *ciphertext* <u>Decryption</u> Input : *ciphertext, IV, key, AAD* Output: *plaintext*	✓		128-, 192-, and 256-bit key sizes Nonce *len*: 7–13	Tag *len*: 4, 6, 8, 10, 12, 14, 16	(Cert # 2299) -AMD Geode (Cert # 2300) -Intel x86	FIPS SP 800-38C	API functions with prefix fips_crypto_ccm
Other non-Security Services							
Initialization Input: *N/A* Output: *N/A*		✓	N/A	N/A	N/A		fips_crypto_module_init
Self Test Input: *N/A* Output: *Return code*		✓	N/A	N/A	N/A		Run_self_test
Get status Input: *N/A* Output: *Module messages*		✓	N/A	N/A	N/A		Kernel log

Table 4: Services

4.3 Operator Authentication

There is no operator authentication; assumption of role is implicit by action.

4.4 Mechanism and Authentication Strength

No authentication is required at security level 1; authentication is implicit by assumption of the role.

5 Physical Security

This is a software module and provides no physical security.

6 Operational Environment

The Module operates in a modifiable operational environment.

6.1 Policy

The Module prevents access by other processes to keys and CSPs during the time the cryptographic module is in the Approved mode. The Module provides a private context per process for key and CSP storage, which is then destroyed upon request by the process or when the Module is powered off. The application that uses the Module is the single user of the Module. No concurrent operators are allowed.

The ptrace(2) system call, the debugger (gdb(1)) and strace(1) shall not be used. In addition, other tracing mechanisms offered by the Linux environment such as ftrace or systemtap shall not be used.

7 Cryptographic Key Management

7.1 Key/CSP Generation

The Module neither generates keys in general nor performs key generation for any of its approved algorithms; instead, keys are passed in from clients by way of algorithm APIs.

7.2 Key Entry and Output

All CSPs enter the Module's logical boundary as cryptographic algorithm API parameters in plaintext. They are associated with memory locations and do not persist across power cycles. The Module does not output intermediate key generation values or other CSPs.

7.3 Key Storage

The Module does not provide persistent key storage for keys or CSPs and they also are not stored inside the Module. Instead, pointers to plaintext keys are passed through the Module and keys/CSPs exist only in the volatile memory that is assigned to the process within which the Module runs.

7.4 Key Zeroization

Whenever CSPs are de-allocated, zeroization is done using different kernel memory zeroization APIs, with a value of 0 and a size equal to that of the CSP. The APIs listed in the table below internally call memset()function for performing zeroization. Table 5 summarizes details regarding what key management the Module provides.

Key/CSP Name	**Details**
128-, 192-, and 256-bit AES keys	Authentication Roles: User, Crypto Officer Generation: N/A Type: Encrypt and decrypt Entry: API parameter Output: N/A Storage: N/A Zeroization API: fips_crypto_free_tfm()
TDES 3-Key	Authentication Roles: User, Crypto Officer Generation: N/A Type: Encrypt and decrypt Entry: API parameter Output: N/A Storage: N/A Zeroization API: fips_crypto_free_tfm()
HMAC keys	Authentication Roles: User, Crypto Officer Generation: N/A

Key/CSP Name	Details
	Type: Keyed-Hash Message Authentication Entry: API function Output: N/A Storage: N/A Zeroization API: fips_crypto_free_ahash()
HMAC key for Module integrity check	Authentication Roles: Crypto Officer Generation: N/A Type: Keyed-Hash Message Authentication Entry: API function Output: N/A Storage: module binary Zeroization: zeroization is not required per FIPS IG 7.4.

Table 5: Key Management Details

8 Electromagnetic Interference/Compatibility

The Module's electromagnetic interference (EMI) and electromagnetic compatibility (EMC) features are summarized in Table 6: EMI and EMC

.

Testing Platform	Product Name/Model	Model Number	EMI/EMC Notes
oMG	oMG	2000	Compliant to FCC part 15 Class A per FCC report
Dell	PowerEdge	R210	Compliant to FCC part 15 Class A per “PowerEdge R210 Dell Technical Guide”

Table 6: EMI and EMC

9 Self Tests

The Module includes known-answer tests that are invoked when the Module is loaded into the kernel. If the known-answer tests fail, error messages are logged in the kernel log file and the Module causes a kernel panic that prevents it from performing further functions. The operating system will be rebooted to recover from the ERROR state. If the tests pass, the file /sys/kernel/crypto_module/fips_initialized will then contain a "1", which indicates the Module is in FIPS mode. The directory /proc/crypto-fips provides a list of the approved algorithms.

9.1 Integrity test

During the software build process, the Module is used to compute a HMAC-SHA-1 message authentication code (MAC) of the Module binary—the MAC and the required key are then stored with the Module. Prior to loading the Module, a HMAC-SHA-1 MAC of the binary is again computed and compared to the original. If the comparison passes, the Module is loaded and the Power-up Tests are run; if the tests pass, the Module enters the FIPS Approved mode. If the comparison fails, the Module is not loaded and is unavailable.

9.2 Power-up Tests

At module start-up, known-answer tests (also referred to as cryptographic algorithm tests)—which are based on the following algorithms—are performed automatically without requiring operator intervention. When the module is performing self tests, no API functions are available and no data output is possible until the module has completed performing the self test. If the value calculated and the known answer do not match, the Module causes a kernel panic.

- AES encryption and decryption are tested separately for ECB, CBC, CTR, GCM and CCM modes
- Triple-DES encryption and decryption are tested separately for ECB and CBC modes
- 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

9.3 On-demand Tests

Self tests may be invoked by restarting the operating system causing the power-up tests to run.

10 Design Assurance

10.1 Configuration Management

The source code for the Module is stored on a server that is connected to a private corporate intranet. Changes to the source code, and other required files, are managed with the git distributed version control system, which provides traceability between developers, the source code, and the released binary module. Each binary is tracked with an embedded build number that has a matching tag in the revision control system, which identifies the source files that were used to produce the binary.

10.2 Delivery and Operation

This module is delivered as a kernel module that is loaded into the kernel after an integrity check is performed. During the kernel module initialization process, the module invokes the Self Tests and upon success, enters FIPS mode. The module is then loaded into the kernel before any client can request the cryptographic services it provides.

11 Mitigation of Other Attacks

No other attacks are mitigated.

12 Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CVT	Component Verification Testing
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
FSM	Finite State Model
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
O/S	Operating System
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SVT	Scenario Verification Testing
TDES	Triple DES

13 References

[1] FIPS 140-2 Standard, <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

[2] FIPS 140-2 Implementation Guidance, <http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf>

[3] FIPS 140-2 Derived Test Requirements, <http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402DTR.pdf>

[4] FIPS 197 Advanced Encryption Standard, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

[5] FIPS 180-4 Secure Hash Standard, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>

[6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), <http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf>

[7] NIST SP 800-38C Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, <http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf>

[8] NIST SP 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf<http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf>