RSA Security, Inc.

RSA™ BSAFE®
Crypto-C Micro Edition Toolkit Module

FIPS 140-2 Certification
Security Policy

Version 1.0.1

Level 1 Validation

August, 2002

Table of Contents

1 INTRODUCTION

1.1 Purpose

This is a non-proprietary cryptographic module security policy for RSA Security, Inc.'s RSA BSAFE Crypto-C ME Toolkit Module version 1.7 (Crypto-C ME Module). This security policy describes how the Crypto-C ME Module meets the security requirements of FIPS 140-2, and how to securely operate the Crypto-C ME Module in a FIPS compliant manner. This policy was prepared as part of the level 1 FIPS 140-2 validation of the Crypto-C ME 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 NIST website at http://csrc.nist.gov/cryptval/.

1.2 References

This document deals only with operations and capabilities of the Crypto-C ME Module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the Crypto-C ME Module and the entire RSA BSAFE product line from the following resources:

- The RSA website contains information on their full line of products and services at http://www.rsa.com.
- An overview of the Crypto-C ME Module is located at http://www.rsasecurity.com/products/bsafe/cryptoc-micro.html.
- The RSA BSAFE product overview is provided at http://www.rsasecurity.com/products/bsafe/index.html.
- For answers to technical or sales related questions please refer to http://www.rsasecurity.com/contact/.

1.3 Terminology

In this document the Crypto-C ME Module will sometimes be referred to as the module.

1.4 Document Organization

The Security Policy document is one document in complete FIPS 140-2 Submission Package. In addition to this document, the complete Submission Package contains:

- Executive summary
- Finite state machine
- Vendor evidence document
- Module software listing
- Developer's Guide
- API documentation
- Other supporting documentation as additional references

This document explains the Crypto-C ME Module's FIPS 140-2 relevant features and functionality. The first section of this document provides an overview and introduction to the

Security Policy. Section 2 describes the Crypto-C ME Module, and how it meets FIPS 140-2 requirements. Section 3 specifically addresses the required configuration for the FIPS-mode of operation.

This Security Policy and other Certification Submission Documentation was produced by Corsec Security, Inc. under contract to RSA Security, Inc. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Certification Submission Documentation is RSA Security, Inc.-proprietary and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact RSA Security, Inc.

2 Crypto-C ME Module

More than one half billion copies of the RSA BSAFE technology are embedded in today's most popular software applications and hardware devices. Encompassing the most widely-used and richest sets of cryptographic algorithms and secure communications protocols, RSA BSAFE software is a set of complementary security products relied-upon by developers and manufacturers worldwide.

The Crypto-C ME Module is RSA Security, Inc.'s cryptographic library designed for securing mobile devices like wireless phones and personal digital assistants. Crypto-C ME's provides performance, interoperability, and flexibility while addressing the significant memory constraints of wireless and embedded devices. It contains assembly-level optimizations on key wireless processors while offering great flexibility and choice by allowing developers to select only the algorithms needed in reduced code sizes. Its functionality includes a wide range of data encryption and signing algorithms, including Triple-DES, the high-performing RC5, the RSA Public Key Cryptosystem, the DSA government signature algorithm, MD5 and SHA-1 message digest routines, and more. With Crypto-C ME developers can use the same secure foundation for the wireless and embedded worlds that RSA Security, Inc. has built in the wired world.

2.1 Cryptographic Module

The Crypto-C ME Module is classified as a multi-chip standalone module for FIPS 140-2 purposes. As such, the module must be evaluated upon a particular operating system and computer platform. The cryptographic boundary thus includes the Crypto-C ME Module running upon an IBM-compatible Personal Computer (PC) running the Microsoft Windows™ 2000 Operating System (OS) while configured in "single user" mode. The Crypto-C ME Module running on this platform was validated as meeting all FIPS 140-2 level 1 security requirements, including cryptographic key management and operating system requirements. The Crypto-C ME Module is packaged in a single dynamic link library (DLL) file, cryptocme2.dll, which contains all the module's executable code.

2.2 Module Interfaces

As a multi-chip standalone module being evaluated on an IBM-compatible PC, the Crypto-C ME Module's physical interfaces consist of the keyboard, mouse, monitor, CD ROM drive, floppy drive, serial ports, USB ports, COM ports, and network adapter(s). However, the module sends/receives data entirely through the underlying logical interface, a C-language Application Program Interface (API) documented in the *RSA BSAFE Crypto-C Mirco Edition API Documentation*. The module provides for Control Input through the API calls. Data Input and

Output are provided in the variables passed with API calls, and Status Output is provided through the returns codes that are documented for each call.

2.3 Roles and Services

The Crypto-C ME Module meets all FIPS140-2 level 1 requirements for Roles and Services, implementing both a User role and Crypto-Officer (CO) role. As allowed by FIPS 140-2, the Crypto-C ME Module does not support user identification or authentication for these roles. Only one role may be active at a time and the Crypto-C ME Module does not allow concurrent operators.

At the highest level, the services provided by the module include:

- Initialization, Operating Controls, and Roles services
- Self-Test and Library Controls services
- Crypto Commands
- Low-Level Random Routines
- Public Key Operations
- Symmetric Key Operations

A complete list of all module services is provided in Appendix A – Services.

2.3.1 Crypto Officer Role

An operator assuming the Crypto Officer role can call any of the module's functions. The complete list of the functionality available to the Crypto Officer is outlined in Appendix A.

2.3.2 User Role

An operator assuming the User role can utilize the entire Crypto-C ME API except for the `me_startup_nist_self_test()` method, which is reserved for the CO. The Crypto-C ME API functions are documented in Appendix A.

2.4 Cryptographic Key Management

2.4.1 Key Generation

The Crypto-C ME Module supports generation of the DSA, RSA, and Diffie-Hellman (DH) public and private keys. Furthermore, the module employs a FIPS 186-2 random number generator using SHA-1 for generating symmetric keys used in algorithms such as AES, DES, or TDES.

2.4.2 Key Storage

The Crypto-C ME Module does not provide long-term cryptographic key storage. If an operator chooses to store keys, the operator is responsible for storing keys exported from the module.

2.4.1 Key Access

Because the Crypto-C ME Module loads into memory with no key material, all key material must either be entered by the an operator, or be generated by the module at the operator's request; hence, an operator has access to all key data created during the module's operation.

2.4.2 *Key Protection/Zeroization*

All key data resides in internally allocated data structures and can only be output using the module's defined API. Microsoft Windows 2000 protects the module's memory and process space from unauthorized access. The operator should follow the steps outline in the *RSA BSAFE Crypto-C Mirco Edition API Documentation* and the *RSA BSAFE Crypto-C ME Developer's Guide* to ensure sensitive data is protected by freeing the data from memory when it is no longer needed.

2.5 Cryptographic Algorithms

The Crypto-C ME Module supports a wide variety of cryptographic algorithms. FIPS 140-2 requires that FIPS-approved algorithms be used whenever there is an applicable FIPS standard. Thus, as the following table summarizes, only a subset of the algorithms provided by the Crypto-C ME Module may be used in compliance with the FIPS 140-2 requirements. For more information on using Crypto-C ME in a FIPS compliant manner refer to Section 3.

Type	Algorithm	FIPS-Approved
Public Key	Diffie-Hellman	No[1]
	DSA (key sizes: 512-1024)	Yes (FIPS 186-2)
	RSA (key sizes: 512-8192)	Yes (FIPS 186-2)
	RSA (enc/dec: 512-8192)	No
Symmetric Key	AES (CBC, CFB, ECB, OFB)	Yes (FIPS 197)
	DES (CBC, CFB, ECB, OFB)	Yes (FIPS 46-3)
	RC2 (CBC, CFB, ECB, OFB)	No
	RC4 (CBC, CFB, ECB, OFB)	No
	RC5 (CBC, CFB, ECB, OFB)	No
	TDES (CBC, CFB, ECB, OFB)	Yes (FIPS 46-3)
Digest	MD2	No
	MD5	No
	SHA-1	Yes (FIPS 180-1)
	SHA-2 (256, 384, 512)	No
MAC	SHA-1 HMAC	Yes (FIPS 198a)
	MD5 HMAC	No
PRNG	FIPS 186-2	Yes (FIPS 186-2)

Table 1 – Algorithms supported by the Crypto-C ME Module

2.6 Self-Test

The Crypto-C ME Module performs a number of power-up and conditional self-tests to ensure proper operation.

2.6.1 *Power-Up Self-Tests*

The power-up self-tests implemented in the Crypto-C ME Module include known answer tests (KAT) for AES, DES, TDES, SHA-1, DSA, and RSA. Also executed at power-up is a software/firmware integrity check. Power-up self-tests are executed automatically when the module is loaded into memory.

[1] DH is not a FIPS approved algorithm but is allowed for use in FIPS-mode as it is a commercially available public-key based key distribution technique.

2.6.2 *Conditional Self-Tests*

The Crypto-C ME Module performs two conditional self-tests: a pair-wise consistency test each time the module generates a DSA, DH, or RSA public/private key pair, and a continuous random number generator test each time the module produces random data per its FIPS 186-2 random number generator.

3 Secure Operation of the Crypto-C ME Module

The Crypto-C ME Module may be placed into FIPS mode by calling the `R_CR_SP_enable_nist_operating_mode` function. After making the `R_CR_SP_enable_nist_operating_mode` function call, the module enforces that only the FIPS approved algorithms outlined in Table 1 are available to operators. To disable FIPS mode, an operator can call `R_CR_SP_enable_non_nist_operating_mode`.

4 Appendix A – Services

This appendix contains a list of the functions provided by the module. For more information see the *FIPS Specific Module Operations* document and the *RSA BSAFE Crypto-C Mirco Edition API Documentation*.

4.1 Initialization

Calls to initialize library and pull in desired cryptographic features:

R_CR_SP_library_init
R_CR_SP_library_free

4.2 Operating Controls

Operating modes control and check whether the module is in FIPS mode:

R_CR_SP_disable_operating_modes
R_CR_SP_operating_mode_is_disabled
R_CR_SP_operating_mode_is_nist
R_CR_SP_enable_nist_operating_mode
R_CR_SP_enable_non_nist_operating_mode
R_CR_SP_operating_mode_is_non_nist

4.3 Roles

These procedures control and check the status of the assumed roles:

R_CR_SP_sign_in_state_is_officer
R_CR_SP_sign_in_state_is_user
R_CR_SP_sign_in_state_is_disabled
R_CR_SP_set_officer_sign_in_state
R_CR_SP_set_user_sign_in_state

4.4 Self-Test

These services allow the CO to initiate and check the results of the power-up self-tests:

R_CR_SP_me_nist_self_test (Note: this function is available to the Crypto Officer only)
R_CR_SP_get_self_test_result

4.5 Library Controls

The `R_CR_SP_get_version` function returns the .DLL's version (as distinct from the version of the underlying crypto toolkit). The `CRYPTOC_ME_library_info` procedure obtains information about the statically linked CryptoC ME library that resides in this module.

4.6 DSA Parameter Control

Services to facilitate algorithm testing.

DSA_generate_parameters

DSA_enable_default_method
DSA_free
DSA_is_prime

4.7 Crypto Commands

The following exports are part of the CryptoC ME API:

R_CR_CTX_get_info
R_CR_CTX_set_info
R_CR_new
R_CR_dup
R_CR_free
R_CR_encrypt_init
R_CR_encrypt
R_CR_encrypt_update
R_CR_encrypt_final
R_CR_decrypt_init
R_CR_decrypt
R_CR_decrypt_update
R_CR_decrypt_final
R_CR_sign_init
R_CR_sign
R_CR_sign_update
R_CR_sign_final
R_CR_verify_init
R_CR_verify
R_CR_verify_update
R_CR_verify_final
R_CR_asym_encrypt_init
R_CR_asym_encrypt
R_CR_asym_decrypt_init
R_CR_asym_decrypt
R_CR_digest_init
R_CR_digest
R_CR_digest_update
R_CR_digest_final
R_CR_key_exchange_init
R_CR_key_exchange_phase_1
R_CR_key_exchange_phase_2
R_CR_generate_key_init
R_CR_generate_key
R_CR_generate_parameter_init
R_CR_generate_parameter
R_CR_random_seed
R_CR_random_bytes
R_CR_get_error_string

R_CR_get_info
R_CR_set_info
R_CR_get_default_method
R_CR_mac_init
R_CR_mac
R_CR_mac_update
R_CR_mac_final
R_CR_verify_mac_init
R_CR_verify_mac
R_CR_verify_mac_update
R_CR_verify_mac_final

4.8 Low-Level Random Routines

Low-Level Random Routines for Testing:

BN_rand
R_rand_meth_sha1
R_rand_add_entropy
R_rand_bytes
R_rand_entropy_count
R_rand_get_default
R_rand_get_entropy_func
R_rand_lib_cleanup
R_rand_load_file
R_rand_seed
R_rand_set_default
R_rand_set_entropy_func
R_rand_write_file

4.9 Public Key Operations

Services used to create and destroy public key objects.

R_PKEY_new
R_PKEY_free
R_PKEY_get_info
R_PKEY_set_info
R_PKEY_CTX_free
R_PKEY_CTX_new

4.10 Symmetric Key Operations

Services used to create and destroy symmetric key objects.

R_SKEY_new
R_SKEY_free

5 Appendix B - Acronym List

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher-block Chaining
CFB	Cipher Feedback
CO	Crypto Officer
DES	Data Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
FSM	Finite State Machine
KAT	Known Answer Test
MD2	Message Digest Algorithm 2
MD5	Message Digest Algorithm 5
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OS	Operating System
PC	Personal Computer
RC2	Rivest's Code 2
RC4	Rivest's Code 4
RC5	Rivest's Code 5
RSA	Rivest, Shamir and Adleman
SHA-1	Secure Hash Algorithm