RSA Security, Inc.

RSA™ BSAFE®
Crypto-J

FIPS 140-1 Validation
Security Policy

Level 1 Validation

December, 2004

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-J Toolkit Module – Version 3.3.4.2 (Crypto-J Module). This security policy describes how the Crypto-J Module meets the security requirements of FIPS 140-1, and how to securely operate the Crypto-J Module. This policy was prepared as part of the level 1 FIPS 140-1 validation of the Crypto-J Module.

FIPS 140-1 (Federal Information Processing Standards Publication 140-1 — *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-1 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-J Module in the technical terms of a FIPS 140-1 cryptographic module security policy. More information is available on the Crypto-J Module and the entire RSA BSAFE product line:

- The RSA website contains information on their full line of products and services at http://www.rsa.com.
- An overview of the Crypto-J Module is located at http://www.rsasecurity.com/products/bsafe/cryptoj.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-J Module will sometimes be referred to as the module.

1.4 Document Organization

The Security Policy document is one document in complete FIPS 140-1 Submission Package. In addition to this document, the complete Submission Package contains:

- Executive summary
- Vendor evidence document
- Finite state machine
- Module software listing
- Other supporting documentation as additional references

This document explains the Crypto-J Module's FIPS 140-1 relevant features and functionality. The first section of this document provides an overview and introduction to the Security Policy. Section 2 describes the Crypto-J Module, and how it meets FIPS 140-1 requirements. Section 3 specifically addresses the required configuration for the FIPS-mode of operation.

This Security Policy and other Validation 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 140-1 Validation 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-J 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-J Module is the world's most trusted Java-language cryptography component and is at the heart of the RSA BSAFE product line. It includes a wide range of data encryption and signing algorithms, including AES, the high-performing RC5, the RSA Public Key Cryptosystem, the DSA government signature algorithm, MD5 and SHA1 message digest routines, and more. Its software libraries, sample code and complete standards-based implementation enable near-universal interoperability for your networked and e-business applications. Any programmer using the RSA BSAFE Crypto-J tools can easily create secure applications without a background in cryptography, mathematics or number theory.

2.1 Cryptographic Module

The Crypto-J Module is classified as a multi-chip standalone module for FIPS 140-1 purposes. As such, the module must be tested upon a particular operating system and computer platform. The cryptographic boundary thus includes the Crypto-J Module running upon an IBM-compatible Personal Computer (PC) running the Windows™ NT Service Pack 6 Operating System (OS) when configured in "single user" mode and using the Java Runtime Environment (JRE) version 1.3.1. The Crypto-J Module running on this platform was validated as meeting all FIPS 140-1 level 1 security requirements, including physical security and operating system requirements. The Crypto-J Module is packaged in a single Java ARchive (JAR) file, jsafeFIPS.jar, which contains all the code for the module.

2.2 Module Interfaces

As a multi-chip standalone module, the Crypto-J Module's physical interfaces consist of the keyboard, mouse, monitor, serial ports, network adapters, etc. However, the underlying logical interface to the module is the Java-language Application Program Interface (API) documented in the *RSA BSAFE Crypto-J Reference Manual*. 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 in the returns and error codes that are documented for each call.

2.3 Roles and Services

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

Because the Crypto-J Module does not require identification or authentication, the module cannot be relied upon to enforce the use of roles and services. The roles and their respective services are therefore defined below.

2.3.1 Crypto Officer Role

An operator assuming the Crypto-Officer role has the ability to start the power-up self-tests on demand by calling the `com.rsa.jsafe.CryptoJ.main()` method with the String argument "verify". The CO can perform this operation manually by going to the command prompt, navigating to the directory containing the jsafeFIPS.jar file, and typing:

```
java –classpath jsafeFIPS.jar com.rsa.jsafe.CryptoJ –verify
```

or programmatically by executing:

```
String[] myArgs = new String[1];
myArgs[0] = "–verify";
com.rsa.jsafe.CryptoJ.main(myArgs);
```

Either of the above commands will pass the string argument "verify" to the `main()` method in `com.rsa.jsafe.CryptoJ.class`. Passing the string "verify" to the `main()` method will invoke the `com.rsa.jsafe.CryptoJ.doVerify()` method which will execute the self-tests. The `com.rsa.jsafe.CryptoJ.main()` is the only function available to the CO.

2.3.2 User Role

An operator assuming the User role can utilize the entire Crypto-J API except for the `com.rsa.jsafe.CryptoJ.main()` method, which is reserved for the CO. The Crypto-J API and its functions and capabilities are documented in the *RSA BSAFE Crypto-J Reference Manual*.

2.4 Physical Security

The Crypto-J Module is a software module tested for use with the Microsoft Windows NT SP6 OS operated in single user mode, but will operate under Microsoft Window 95, 98, 2000, and XP, Linux, Solaris and other UNIX variants. The module was tested against FIPS 140-1 requirements on a standard Intel platform PC running Windows NT SP6 that meets all FIPS 140-1 level 1 physical requirements. This platform provides production grade equipment, industry-standard passivation, and a strong enclosure.

Although the Crypto-J Module consists entirely of software, the FIPS 140-1 tested platform is a standard PC which has been tested for and meets applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined in Subpart B of FCC Part 15.

2.5 Cryptographic Key Management

2.5.1 Key Generation

The Crypto-J Module employs two FIPS-approved random number generation methods, one specified in the FIPS 186-2 and one in the ANSI X9.31 standards. The module will automatically use the FIPS 186-2 compliant random number generator when generating DSA public and private keys.

Additionally to remain in FIPS mode, the User must only use the ANSI X9.31 compliant random number generator for generation of key material including RSA and DH public/private key pairs and any symmetric keys.

2.5.2 Key Storage

The Crypto-J Module does not provide long-term cryptographic key storage. If a User chooses to store keys, the User is responsible for storing keys exported from the module.

2.5.3 *Key Protection*

All key data resides in internally allocated data structures and can only be output using the module's defined API. Microsoft Windows NT protects memory and process space from unauthorized access. Additional steps can be taken by a User to ensure sensitive data is protected by making use of the module's built-in memory obfuscator. Guidelines for using the memory obfuscator are available in the "Memory Obfuscation" section of the *RSA BSAFE Crypto-J Developer's Guide*.

2.5.4 Key Output

The module performs manual key distribution and plaintext electronic key output. Key output is performed by using the getKeyData() function in the JSAFE_Key class. Complete documentation for this function can be found in the *RSA BSAFE Crypto-J Reference Manual.*

2.5.5 Key Zeroization

All key data resides in internally allocated data structures that are "cleaned up" by the Java Virtual Machine's (JVM) garbage collector. Because Java often handles memory in ways that are unpredictable and transparent to the User, a User can take additional steps to ensure sensitive data is zeroized by making use of the module's functions for clearing sensitive data. Guidelines for clearing sensitive data are available in the "Clearing Sensitive Data" section of the *RSA BSAFE Crypto-J Developer's Guide*.

2.5.6 Critical Security Parameter

The module only has one critical security parameter. It is the public key used to verify the modules integrity at power-up. This key is embedded within the jsafeFIPS.jar file.

2.6 Cryptographic Algorithms

The Crypto-J Module supports a wide variety of cryptographic algorithms. FIPS 140-1 requires that FIPS-defined algorithms be used whenever there is an applicable FIPS standard when the module is operated in FIPS mode. Thus, as the following table summarizes, only a subset of the

algorithms provided by the Crypto-J Module may be used in compliance with FIPS 140-1 requirements. For more information on the FIPS 140-1 Mode, please refer to Section 3.

Table 1 – Algorithms supported by the Crypto-J Module

Type	Algorithm	FIPS
Public-Key Algorithms	The RSA Public-Key Cryptosystem	FIPS 186-2
	Diffie-Hellman Key Agreement	Allowed for use in FIPS Mode
	Digital Signature Algorithm (DSA)	FIPS 186-2
Symmetric-Key Algorithms	DES (ECB, CBC, CFB, and OFB)	FIPS 46-3
	DESX	
	Triple DES (ECB, CBC, CFB, and OFB)	FIPS 46-3
	The RC2® block cipher	
	The RC4® stream cipher	
	The RC5® block cipher	
	AES (ECB, CBC, CFB128, OFB)	FIPS 197
	MD2	
Message Digests	MD5	
	SHA1	FIPS 180-1
	HMAC SHA-1	FIPS 198
MAC Algorithm	MD5	
Random-Number Generation	Non FIPS-approved method based on SHA1	
	FIPS 186-2 (used internally for DSA key generation)	FIPS 186-2
	ANSI X9.31	ANSI X9.31
	Base64	
Recode Algorithm		

2.7 Self-Test

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

2.7.1 Power-Up Self-Tests

The power-up self-tests implemented in the Crypto-J module include known answer tests for DES, TDES, AES, SHA-1, DSA, and RSA, and a software/firmware integrity check. Power-up self-tests are executed automatically when the module is loaded by the Java Runtime Environment (JRE).

2.7.2 Conditional Self-Tests

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

3 Secure Operation of the Crypto-J Module

The Crypto-J Module does not require any special configuration to operate in conformance with FIPS 140-1 requirements. The module offers a wide array of the most advanced cryptographic algorithms available. FIPS 140-1, however, requires that only FIPS-approved algorithms be used when operating a FIPS 140-1 compliant manner. Therefore, to operate the Crypto-J Module in conformance with FIPS 140-1 requirements, only the FIPS-approved algorithms listed in section 2.6 may be used.

Note: It is the User's responsibility to understand which algorithms are FIPS-approved and which are not. NIST supports a web site (referenced in section 1.2) that lists certified implementations of NIST-approved cryptographic algorithms.

Acronym List

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher-block Chaining
CFB	Cipher Feedback
CO	Crypto Officer
DES	Data Encryption Standard
DESX	Data Encryption Standard Xored
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
HMAC	Hash Message Authentication Code
JAR	Java Archive
JRE	Java Runtime Environment
JVM	Java Virtual Machine
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
RC3	Rivest's Code 3
RC4	Rivest's Code 4
RC5	Rivest's Code 5
RSA	Rivest, Shamir and Adleman
SHA1	Secure Hash Algorithm