RSA BSAFE®

Crypto-J Software Module

Security Policy (jsafe)

Versions 3.5, 3.5.2 and 3.5.3
November 22, 2006

Cryptographic components for Java

RSA SECURITY®

Contact Information

See our Web sites for regional Customer Support telephone and fax numbers.

RSA Security Inc.	RSA Security Ireland Limited
www.rsasecurity.com	www.rsasecurity.ie

Trademarks

License Agreement

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import or export of encryption technologies and current use, import and export regulations should be followed when exporting this product.

Distribution

RSA Security Notice

The RC5® Block Encryption Algorithm With Data-Dependent Rotations is protected by U.S. Patent #5,724,428 and #5,835,600.
Compaq MultiPrime™ technology is protected by U.S. Patent #5,848,159 and is the subject of patent applications in other countries.
This product includes patented technology licensed from Entrust Technologies Inc. (US Patent# 5,699,431).

Published November 22, 2006

Table of Contents

1 Introduction

This is a non-proprietary cryptographic module security policy for the RSA BSAFE Crypto-J Toolkit Module versions 3.5, 3.5.2 and 3.5.3 (Crypto-J Module). This security policy describes how the Crypto-J Module meets the security requirements of FIPS 140-2, and how to securely operate the Crypto-J Module. This policy is prepared as part of the Level 1 FIPS 140-2 validation of the Crypto-J Module.

The Crypto-J distribution includes two API interfaces:

- jsafeFIPS.jar "JSAFE" application programmer interface to the Crypto-J Module
- jsafeJCEFIPS.jar "JCE" application programmer interface to the Crypto-J Module.

This security policy deals only with the JSAFE interface to the Crypto-J Module. For details of how the FIPS 140-2 evaluation applies to the JCE module, please refer to the document entitled "RSA BSAFE® Crypto-J Security Policy (jsafeJCE)."

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 http://csrc.nist.gov/cryptval/.

1.1 References

This document deals only with operations and capabilities of the Crypto-J Module in the technical terms of a FIPS 140-2 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 the full line of products and services at http://www.rsasecurity.com
- An overview of the Crypto-J Module is located at http://www.rsasecurity.com/node.asp?id=1204
- The RSA BSAFE product overview is provided at http://www.rsasecurity.com/node.asp?id=1202
- For answers to technical or sales related questions, please refer to the contact details in section 6 “Contacting RSA Security” on page 16.

1.2 Terminology

The Crypto-J Module is also referred to as the Cryptographic Module, and as the module.

1.3 Document Organization

This security policy document is one of the documents in the complete FIPS 140-2 Certification 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 Cryptographic Module's features and functionality relevant to FIPS 140-2. This section, "Introduction", provides an overview and introduction to the Security Policy. "Crypto-J Module" on page 6 describes the Cryptographic Module and how it meets the FIPS 140-2 requirements. "Secure Operation of the Crypto-J Module" on page 12 specifically addresses the required configuration for the FIPS-mode of operation. "Services" on page 13 lists all of the functions provided by the Cryptographic Module. "Acronyms" on page 14 lists the definitions for the acronyms used in this document.

With the exception of this non-proprietary security policy, the FIPS 140-2 Certification Submission Documentation is RSA Security-proprietary, and releasable only under appropriate non-disclosure agreements. For access to these documents, please contact RSA Security.

2 Crypto-J Module

This section provides an overview of the Crypto-J Module, through the following topics:

- Introduction
- Cryptographic Module
- Module Interfaces
- Roles and Services
- Cryptographic Key Management
- Cryptographic Algorithms
- Self-Test.

2.1 Introduction

More than half a 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 rich set of cryptographic algorithms as well as secure communications protocols, RSA BSAFE software is a set of complementary security products relied on 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 Triple-DES, 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.

2.2 Cryptographic Module

This Cryptographic Module is classified as a multi-chip standalone module for FIPS 140-2 purposes. As such, the module is tested on a particular operating system and computer platform. The cryptographic boundary includes the Cryptographic Module running on selected platforms that are running selected operating systems, while configured in "single user" mode.

The Cryptographic Module is validated for all FIPS 140-2 Level 1 security requirements, including cryptographic key management and operating system requirements. The Crypto-J Module is packaged in a Java Archive (JAR) file containing all the code for the module. In addition, the Crypto-J Module relies on the physical security provided by the host PC in which it runs.

The JSAFE application programmer interface to the Crypto-J Module is provided in the jsafeFIPS.jar file.

The Crypto-J Module is tested on the following platforms:

- Microsoft Windows
    - Microsoft Windows XP (SP2) – Sun JDK 1.4.2 (32-bit).

Compliance is maintained on platforms for which the binary executable remains unchanged. This includes (but is not limited to):

- Microsoft Windows
    - Microsoft Windows NT 4 Sun JDK 1.1.8/1.3.1/1.4.2/1.5, IBM JDK 1.4.2
    - Microsoft Windows 2000 Service Pack 4 Sun JDK 1.1.8/1.3.1/1.4.2/1.5, IBM JDK 1.4.2
    - Microsoft Windows XP (SP1) Sun JDK 1.1.8/1.3.1/1.4.2/1.5, IBM JDK 1.4.2
    - Microsoft Windows 2003 Server. Sun JDK 1.1.8/1.3.1/1.4.2/1.5, IBM JDK 1.4.2.
- Sun Microsystems
    - Sun Microsystems Solaris 8, Sparc v9 (32-bit) Sun JDK 1.3.1/1.4.2/1.5(32-bit)
    - Sun Microsystems Solaris 8, Sparc v9 (64-bit) Sun JDK 1.5 (64-bit)
    - Sun Microsystems Solaris 9, Sparc v9 (32-bit) Sun JDK 1.3.1/1.4.2/1.5(32-bit)
    - Sun Microsystems Solaris 9, Sparc v9 (64-bit) Sun JDK 1.5 (64-bit)
    - Sun Microsystems Solaris 10, Sparc v9 (32-bit) Sun JDK 1.3.1/1.4.2/1.5(32-bit)
    - Sun Microsystems Solaris 10, Sparc v9 (64-bit) Sun JDK 1.5 (64-bit)
    - Sun Microsystems Solaris 10, Intel x86 (32-bit) Sun JDK 1.3.1/1.4.2/1.5(32-bit).
- Red Hat Linux 7.2 x86 (32-bit)
    - Red Hat Linux 7.2 x86 (32-bit) Sun JDK 1.3.1/1.4.2/1.5
    - Red Hat Advanced Server 3.0 x86 (32-bit). Sun JDK 1.3.1/1.4.2/1.5.
- HP-UX 11 (32-bit)
    - HP-UX 11.0 PA-RISC 1.1 (32-bit) HP JDK 1.4.2
    - HP-UX 11.0 PA-RISC 2.0 (32-bit) HP JDK 1.4.2
    - HP-UX 11.22 Itanium2 (32-bit) HP JDK 1.4.2
    - HP-UX 11.23 Itanium2 (32-bit) HP JDK 1.4.2.
- IBM AIX 5L v5.3 (32bit)
    - IBM AIX 5L v5.3, PowerPC (32-bit) IBM JDK 1.4.2.

For a resolution on the issue of "Multi user" modes, see the NIST document *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program* at the government website http://csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf.

2.3 Module Interfaces

As a multi-chip standalone module, the Crypto-J Module's physical interfaces consist of a keyboard, mouse, monitor, serial ports, network adapters, and so on.

The underlying logical interface to the module is the Application Program Interface (API), documented in the *RSA BSAFE Crypto-J Developer's Guide*. 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 documented for each call.

2.4 Roles and Services

The Crypto-J Module meets all FIPS140-2 Level 1 requirements for Roles and Services, implementing both a Crypto-Officer role and a User role. As allowed by FIPS 140-2. the module does not require user identification or authentication for these roles. Only one role can be active at a time, and the module does not allow concurrent operators.

The API for control of the module are through the "CryptoJ" class.

JAR File	Prefix
jsafeFIPS.jar	com.rsa.jsafe

2.4.1 Crypto-Officer Role

An operator can assume the Crypto-Officer role by invoking the method `<PREFIX>.CryptoJ.setRole()` with the argument CRYPTO_OFFICER_ROLE.

Once in the Crypto-Officer role, the operator can start the power-up self-tests on demand by calling the method `<PREFIX>.CryptoJ.runSelfTests()`.

The Crypto-Officer can perform this operation manually at the command prompt by navigating to the directory containing the appropriate jar file, and typing:

```
java -cp <JARFILE> <PREFIX>.CryptoJ -testAll
```

Alternatively, the Crypto-Officer can call the operation programmatically:

```
<PREFIX>.CryptoJ.runSelfTests();
```

When the Crypto-J Module is loaded, the power-up self-tests run automatically. After passing the integrity check, the self-tests will not run again unless the module is unloaded and reloaded. So, calling the self-tests on demand only results in the power-up known-answer tests and pairwise consistency checks being performed.

2.4.2 User Role

The User role is the default operating role. However, an operator can explicitly assume the User role by invoking the method `<PREFIX>.CryptoJ.setRole()` with the argument USER_ROLE.

The Crypto-J Module API, its functions, and capabilities are documented in the *RSA BSAFE Crypto-J Developer's Guide*. A full list of services is also provided in this document on page 13.

2.5 Cryptographic Key Management

2.5.1 Key Generation

The Crypto-J Module supports generation of the DSA, RSA, and Diffie-Hellman (DH) public and private keys. The module also employs a FIPS 186-2 compliant random number generator for generating asymmetric and symmetric keys used in algorithms such as AES, DES[1], TDES, RSA, DSA or Diffie-Hellman.

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 returned to the calling application.

Volatile (that is, short-term) memory storage of cryptographic keys and CSPs employed by the cryptographic module is handled as shown in the following table. The User and CO roles have equal and complete access to all keys and CSPs.

Table 1. Key and CSP storage

Item	Storage
AES keys	In volatile memory only (plaintext)
DES keys	In volatile memory only (plaintext)
Triple DES keys	In volatile memory only (plaintext)
HMAC with SHA1 and SHA2 keys	In volatile memory only (plaintext)
Diffie-Hellman public key	In volatile memory only (plaintext)
Diffie-Hellman private key	In volatile memory only (plaintext)
RSA public key	In volatile memory only (plaintext)
RSA private key	In volatile memory only (plaintext)
DSA public key	In volatile memory only (plaintext)
DSA private key	In volatile memory only (plaintext)
PRNG seeds (FIPS 186-2)	In volatile memory only (plaintext)

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. The operating system and Java Runtime Environment protects memory and process space from unauthorized access.

2.5.4 Key Zeroization

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

[1] For legacy system use only; transitional phase only – valid until May 19th, 2007.

2.6 Cryptographic Algorithms

The Crypto-J Module supports a wide variety of cryptographic algorithms. When the module is operated in FIPS-mode, FIPS 140-2 requires that FIPS-approved algorithms be used. Table 2 lists the algorithms provided by the Crypto-J Module that comply with the FIPS 140-2 requirements. Table 3 lists the non-FIPS-approved algorithms.

Table 2. Crypto-J FIPS-approved algorithms

Algorithm	Certificate Number
AES – ECB, CBC, CFB (128), OFB (128) – [128, 192, 256 bit key sizes]	270
DES - ECB, CBC, CFB (64bit) , and OFB (64 bit) (for legacy use only; transitional phase only – valid until May 19th, 2007)	325
Diffie-Hellman Key Agreement	Non-Approved (allowed in FIPS mode)
Digital Signature Algorithm (DSA)	139
FIPS 186-2 General Purpose (x-Change Notice(SHA-1)	105
HMAC-SHAx (where x is 1, 224, 256, 384, or 512)	85
RSASSA-PSS (sign, verify) (SHA-1)	70 (Crypto-J 3.5 and 3.5.2) 185 (Crypto-J 3.5.3)
RSA PKCS#1 v1.5 (sign/verify) (SHA-1,SHA-224,SHA-256,SHA-384,SHA-512)	70 (Crypto-J 3.5 and 3.5.2) 185 (Crypto-J 3.5.3)
Secure Hash Standard (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)	355
Triple DES - ECB, CBC, CFB (64 bit), and OFB (64 bit)	353
RSA X9.31 (keygen, sign, verify)	70 (Crypto-J 3.5 and 3.5.2) 185 (Crypto-J 3.5.3)

For more information on using Crypto-J in a FIPS compliant manner, see the section "Secure Operation of the Crypto-J Module" on page 12.

Table 3. Crypto-J non-FIPS-approved algorithms

Algorithm
DESX
MD2
MD5
Random Number Generators (ANSI X9.31, MD5Random, SHA1Random)
The RC2® block cipher
The RC4® stream cipher
The RC5® block cipher
PBEWithSHA1And3DES
RSA OAEP for key transport
Raw RSA encryption and decryption
RSA Keypair Generation MultiPrime (2 or 3 primes)
RIPEMD160
HMAC-MD5

2.7 Self-Tests

The Crypto-J Module performs a number of power-up and conditional self-tests to ensure proper operation. If any of these tests fails, the module throws a SecurityException, which provides a status output, and aborts the operation that caused the conditional self-tests to fail.

2.7.1 Power-Up Self-Tests

The power-up self-tests implemented in the Crypto-J module are as follows:

- PRNG KATs
- AES KATs
- DES KATs
- TDES KATs
- SHA-1 KATs
- SHA-224 KATs
- SHA-256 KATs
- SHA-384 KATs
- SHA-512 KATs
- HMAC SHA-1 KATs
- HMAC SHA-224 KATs
- HMAC SHA-256 KATs
- HMAC SHA-384 KATs
- HMAC SHA-512 KATs
- pairwise consistency checks for DSA and RSA
- 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.

2.7.3 Mitigation of Other Attacks

RSA key operations implement blinding by default, providing a defense against timing attacks. Blinding is implemented through blinding modes, for which the following options are available:

- Blinding mode off
- Blinding mode with no update, where the blinding value is constant for each operation
- Blinding mode with full update, where a new blinding value is used for each operation.

3 Secure Operation of the Crypto-J Module

The Crypto-J Module does not require any special configuration to operate in conformance with FIPS 140-2 requirements. The following guidance must be followed, however, to achieve a FIPS-mode of operation.

3.1 Crypto-Officer Guidance

The Crypto-Officer is responsible for installing the module. Installation instructions are provided in the *RSA BSAFE Crypto-J Installation Guide*.

The module's default state is FIPS_MODE. Crypto-Officers must not take the module out of FIPS_MODE.

3.2 User Guidance

The User must only use algorithms approved for use in a FIPS-mode of operation. Table 2 lists the approved algorithms. The FIPS-approved bit-length for a DSA key pair must be between 512 and 1024 bits in multiples of 64, and the FIPS-approved RNGs must be seeded with values of at least 160 bits in length. The FIPS-approved bit lengths for an RSA[2] key pair must be between 1024 and 4096 bits in multiples of 512. The FIPS-approved bit lengths for the Diffie-Hellman[3] key agreement must be between 1024 and 2048 bits. The FIPS-approved bit lengths for an HMAC key must be between 80 and 4096 bits.

If RSA key generation is requested in FIPS mode, the module always uses the FIPS-approved RSA X9.31 key-generation procedure.

 The module's default state is FIPS_MODE. Users must not take the module out of FIPS_MODE.

Users should take care to zeroize CSPs when they are no longer needed. Please follow the procedure for clearing sensitive data as outlined in the "Clearing Sensitive Data" section of the *RSA BSAFE Crypto-J Developer's Guide*.

[2] When used for transporting keys and using the minimum allowed modulus size, the minimum strength of encryption provided is 80 bits.

[3] Using the minimum allowed modulus size, the minimum strength of encryption provided is 80 bits.

4 Services

The Crypto-J Module meets all FIPS 140-2 Level 1 requirements for Roles and Services, implementing both a Crypto-Officer role and a User role. As allowed by FIPS 140-2, the module does not require user identification or authentication for these roles. Only one role can be active at a time, and the module does not allow concurrent operators.

The following table lists the services provided by the Crypto-J Module in terms of the module's interface. For more information on each function, see the *RSA BSAFE Crypto-J Developer's Guide*.

Table 4. Services for Crypto-J (jsafeFIPS.jar)

Service	Service	Service
CryptoJ.runSelfTests*	JSAFE_KeyAgree	JSAFE_PublicKey
CryptoJ.setRole	JSAFE_KeyAttributes	JSAFE_Recode
CryptoJ.getRole	JSAFE_KeyPair	JSAFE_SecretKey
CryptoJ.setMode	JSAFE_MAC	JSAFE_SecureRandom
CryptoJ.getMode	JSAFE_MessageDigest	JSAFE_Session
CryptoJ.getState	JSAFE_Obfuscator	JSAFE_SessionSpec
CryptoJ.selfTestPassed	JSAFE_Object	JSAFE_Signature
JSAFE_AsymmetricCipher	JSAFE_Parameters	JSAFE_SymmetricCipher
JSAFE_Key	JSAFE_PrivateKey	JSAFE_VerifyPQG

*Only available to the Crypto-Officer role.

5 Acronyms

Acronym	Definition
AES	Advanced Encryption Standard. A fast block cipher with a 128-bit block, and keys of lengths 128, 192 and 256 bits. This will replace DES as the US symmetric encryption standard.
API	Application Programming Interface.
Attack	Either a successful or unsuccessful attempt at breaking part or all of a cryptosystem. Various attack types include an algebraic attack, birthday attack, brute force attack, chosen ciphertext attack, chosen plaintext attack, differential cryptanalysis, known plaintext attack, linear cryptanalysis, and middleperson attack.
CBC	Cipher Block Chaining. A mode of encryption in which each ciphertext depends upon all previous ciphertexts. Changing an IV alters the ciphertext produced by successive encryptions of an identical plaintext.
CFB	Cipher Feedback. A mode of encryption that produces a stream of ciphertext bits rather than a succession of blocks. In other respects, it has similar properties to the CBC mode of operation.
CSP	Cryptographic Service Provider.
DES	Data Encryption Standard. A symmetric encryption algorithm with a 56-bit key. See also Triple DES.
Diffie-Hellman	The Diffie-Hellman asymmetric key exchange algorithm. There are many variants, but typically two entities exchange some public information (for example, public keys or random values) and combines them with their own private keys to generate a shared session key. As private keys are not transmitted, eavesdroppers are not privy to all of the information that composes the session key.
DSA	Digital Signature Algorithm. An asymmetric algorithm for creating digital signatures.
ECB	Electronic Code Book. A mode of encryption in which identical plaintexts are encrypted to identical ciphertexts, given the same key.
Encryption	The transformation of plaintext into an apparently less readable form (called ciphertext) through a mathematical process. The ciphertext may be read by anyone who has the key that decrypts (undoes the encryption) the ciphertext.
FIPS	Federal Information Processing Standards.
HMAC	Keyed-Hashing for Message Authentication Code.
KAT	Known Answer Test.
Key	A string of bits used in cryptography, allowing people to encrypt and decrypt data. Can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext. Various types of keys include: distributed key, private key, public key, secret key, session key, shared key, subkey, symmetric key, and weak key.
MD5	A secure hash algorithm created by Ron Rivest. MD5 hashes an arbitrary-length input into a 16-byte digest.
NIST	National Institute of Standards and Technology. A division of the US Department of Commerce (formerly known as the NBS) which produces security and cryptography-related standards.
OFB	Output Feedback. A mode of encryption in which the cipher is decoupled from its ciphertext.
OS	Operating System.
PC	Personal Computer.
private key	The secret key in public key cryptography. Primarily used for decryption but also used for encryption with digital signatures.
PRNG	Pseudo Random Number Generator.
RC2	Block cipher developed by Ron Rivest as an alternative to the DES. It has a block size of 64 bits and a variable key size. It is a legacy cipher and RC5 should be used in preference.
RC4	Symmetric algorithm designed by Ron Rivest using variable length keys (usually 40 bit or 128 bit).
RC5	Block cipher designed by Ron Rivest. It is parameterizable in its word size, key length and number of rounds. Typical use involves a block size of 64 bits, a key size of 128 bits and either 16 or 20 iterations of its round function.

Acronym	Definition
RNG	Random Number Generator.
RSA	Public key (asymmetric) algorithm providing the ability to encrypt data and create and verify digital signatures. RSA stands for Rivest, Shamir, and Adleman, the developers of the RSA public key cryptosystem.
SHA	Secure Hash Algorithm. An algorithm which creates a unique hash value for each possible input. SHA takes an arbitrary input which is hashed into a 160-bit digest.
SHA-1	A revision to SHA to correct a weakness. It produces 160-bit digests. SHA-1 takes an arbitrary input which is hashed into a 20-byte digest.
SHA-2	The NIST-mandated successor to SHA-1, to complement the Advanced Encryption Standard. It is a family of hash algorithms (SHA-256, SHA-384 and SHA-512) which produce digests of 256, 384 and 512 bits respectively.
TDES	Triple-DES.

6 Contacting RSA Security

The RSA Security Web site contains the latest news, security bulletins and information about coming events.

The RSA BSAFE Web site contains product information.

The RSA Laboratories Web site contains frequently asked questions.

6.1 Support and Service

If you have any questions or require additional information, see RSA Support or RSA SecurCare Online.

6.2 Purchasing Printed Product Documentation

All documentation for your RSA Security product is included in electronic format on the CD or in the download you have received. You can print product documentation directly from these files if you require a hard copy.

RSA Security also offers customers the option to purchase printed and bound copies of key documents for some products. Contact your RSA Security sales representative for more information.

6.3 Feedback

We welcome your feedback on RSA Security documentation. Please email bsafeuserdocs@rsasecurity.com.