[redacted]

Entrust, Inc.

Cryptographic Module Security Policy

Entrust Authority Security Toolkit for Java 7.0

Author: Chris Wood
Date: September 29, 2004
Version: 2.2

Table of Contents

1 Revision History

Authors	Date	Version	Comment
Chris Wood	June 12, 2002	1.0	First Version
Chris Wood	October 15, 2002	1.1	Comments from DOMUS
Chris Wood	March 19, 2003	1.2	Comments from NIST
Chris Wood	March 28, 2004	2.0	Java Toolkit 7.0 Submission
Chris Wood	September 29, 2004	2.1	Comments from NIST
Chris Wood	October 29, 2004	2.2	Algorithm certificates added

Contributors	Topics
Marc Laroche	Suggestions, guidance

2 References

Author	Title
NIST	[1] FIPS PUB 140-2: Security Requirements For Cryptographic Modules, May 2001
NIST	[2] Derived Test Requirements for FIPS PUB 140-2, March 2004
NIST	[3] Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, March 2004
Entrust	[4] Security Toolkit for Java 7.0 - Programmer's Guide, 2004
Entrust	[CMC] Cryptographic Module Classes for the Security Toolkit for Java 7.0, March 2004
Entrust	[CR] Cryptographic Module Validation Cross-Reference for the Security Toolkit fro Java 7.0, March 2004
Entrust	[DD] Cryptographic Module Design Description for the Security Toolkit for Java 7.0, March 2004
Entrust	[FD] Cryptographic Module Functional Description for the Security Toolkit for Java 7.0, March 2004
IBM	[UG] User guide for NetVista 8181, 8182, 8301, 8303, 8304, 8305, 8306, 8307, 8308, 8309, 8310, 8311, 8312, 8313, 8314, and 8315 systems (English), IBM Corporation, 2002, (ftp://ftp.software.ibm.com/pc/pccbbs/netvista_pdf/49p0935.pdf)
IBM	[HMM] Hardware Maintenance Manual Types 8301, 8302, 8303, 8304, 8305, 8306, 8307, 8308, 8309, 8310, 8311, 8312, 8313, 8314, and 8315, IBM Corporation, 2001 (ftp://ftp.software.ibm.com/pc/pccbbs/netvista_pdf/24p2969.pdf)
Sun	[SM] Sun Ultra 10 Service Manual, Sun Microsystems Inc., 2000 (http://www.sun.com/products-n-solutions/hardware/docs/pdf/805-7764-12.pdf)

3 Target Audience

This document is intended to be part of the package of documents that are sent for FIPS validation. It is intended for the following people:

- NIST and the FIPS 140-2 validation group
- CSE
- Developers working on the release
- Product Verification

- Documentation
- Product and Development Managers
- Security Assurance

4 Introduction

This document contains a description of the Entrust Authority™ Security Toolkit for Java™ (JTK) Cryptographic Module Security Policy. It contains a specification of the rules under which the JTK cryptographic module must operate. These security rules were derived from the requirements of FIPS 140-2 [1].

4.1 Purpose of the Security Policy

There are three major reasons that a security policy is defined for and must be followed by the cryptographic module:

- 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, protection, and access rights provided by the cryptographic module, allowing individuals and organizations to determine whether it will meet their security requirements.

4.2 Cryptographic Module Definition

This section defines the cryptographic module that is being submitted for validation to FIPS PUB 140-2, level 1.

The module consists of the following generic components:

1. A commercially available general-purpose hardware-computing platform. A generic high-level block diagram for such a platform is provided in Figure 1.
2. A commercially available Operating System (OS) that runs on the above platform.
3. The Java Runtime Environment.
4. A software component, the JTK (set of '.class' files) that runs on the above platform, operating system, and Java runtime environment. This component is custom designed and written by Entrust in the Java computer language and is identical, at the source code level, for all identified hardware platforms and operating systems. The source code (see [FD] for list of classes) is compiled into Java byte-code for interpretation by the Java Virtual Machine on the above OS or Browser. An Application Programming Interface (API) is defined as the interface to the cryptographic module.

The cryptographic module has two main platform configurations, which are also the FIPS 140-2 test platforms (OS and JRE as noted below). For each configuration the hardware computing platform, operating system, and Java implementation are listed below.:

1. An IBM NetVista 8305-D1U Personal Computer system with:
    - Intel Pentium 4 2.8GHz processor with 512KB L2 cache
    - Intel 845G chipset
    - IBM 1GB PC2100 CL2.5 DDR SDRAM UDIMM
    - IBM 80GB 7200RPM ATA-100 EIDE hard disk drive
    - IBM IDE 48x24x48x16x Max CD-RW/DVD-ROM combo drive
    - Integrated Intel 10/100 Ethernet with (Wake on Lan)
    - 185 watt power supply with variable speed fan
    - Six USB ports (two in front, four in back) [Ver 2.0]
    - One ethernet port RJ-45
    - Two serial 9-pin ports
    - One parallel port(EPP, ECP), IEEE 1284

Operating Systems:
- Windows 2000 Professional, Server and Advanced Server
- Windows Server 2003
- Windows XP Professional (Test Platform OS)

Java Runtime Environments:
- Sun JRE 1.3.1
- Sun JRE 1.4.2 (Test Platform JRE)
- IBM JRE 1.3.1
- IBM JRE 1.4.1

A detailed technical description of the IBM NetVista 8305-D1U platform is included in [UG] and [HMM].

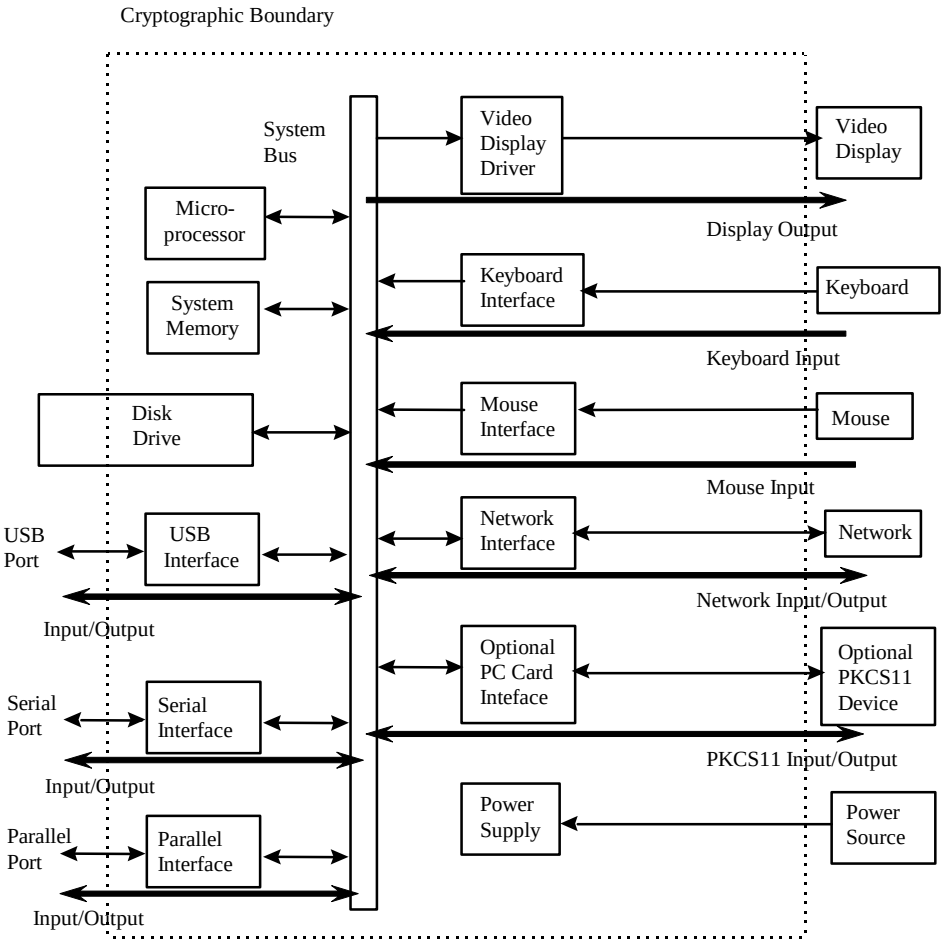
2. A Sun Ultra 10 Workstation with:
   - UltraSPARC-IIi 300 MHz processor with 512KB external cache
   - Up to 1GB of 50ns or 60ns 168-pin JEDEC DIMM RAM
   - Up to two 4.3 GB (4500 rpm) or 9.1/20.4 GB (7200 rpm)
   - One 24X-, 32X, or 48X-speed CD-ROM,
   - One 3.5-in. 1.44-MB floppy
   - One ethernet/fast ethernet (10BASE-T/100BASE-T)
   - 380 watt power supply
   - One ethernet port RJ-45
   - One synchronous/asynchronous serial port
   - One IEEE 1284 (bidirectional) parallel port

   Operating Systems:
   - Solaris 8
   - Solaris 9 (Test Platform OS)

   Java Runtime Environments:
   - Sun JRE 1.3.1
   - Sun JRE 1.4.2 (Test Platform JRE)

A detailed technical description of the Sun Ultra 10 platform is included in [SM].

The JTK cryptographic module is also suitable for platforms from the same or other manufacturers, based on compatible processors with equivalent or greater system resources, equivalent or later Operating System versions, and equivalent or later Java Runtime Environment versions. Also, the JTK cryptographic module used on all Microsoft Operating Systems is identical.



Note: All arrows indicate data flow, however; only bold arrows indicate data (plaintext and encrypted) flows into and out of the Cryptographic Module via physical ports

Figure 1: Cryptographic module block diagram for hardware.

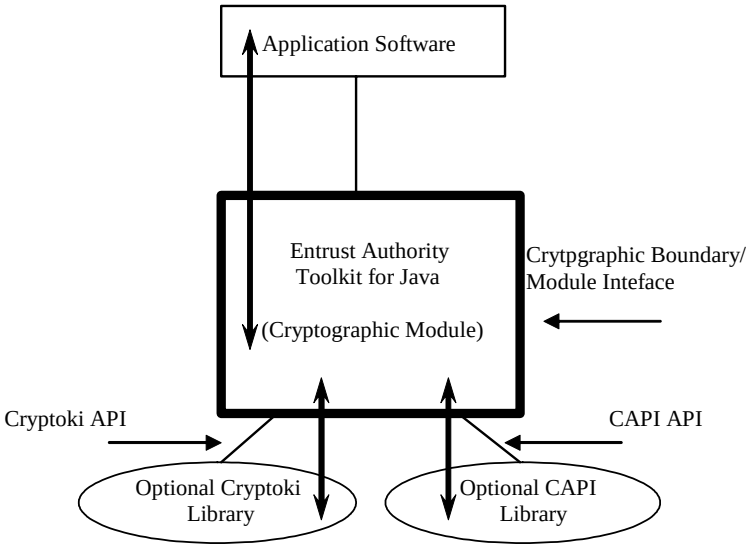


Figure 2: Cryptographic module block diagram for software.

4.3 Cryptographic Module Description

The cryptographic module consists of a defined subset of Java .class files from the JTK. These classes are listed and described in the Cryptographic Module Classes [CMC] companion document. The cryptographic module provides a set of functions (API) that allows developers to integrate the cryptographic module security features into the applications they design. The cryptographic module API is described in detail in the Cryptographic Module Functional Description [FD] companion document.

The purpose of the cryptographic module is to provide application developers with the access to cryptographic algorithms, and the ability to integrate security into the applications they design. The types of cryptographic algorithms provided include:

- Symmetric Ciphers (encryption/decryption/key generation)
- Asymmetric Ciphers (encryption/decryption/key generation)
- Message Digests (hashing)
- Signatures (signing/verification)
- Message Authentication Codes (creation)
- Keyed-Hash Message Authentication Codes (creation)
- Random Number/Seed Generation
- Key Agreement

5 Specification of the Security Policy

5.1 Identification and Authentication Policy

The Cryptographic Module does not identify nor authenticate any user (in any role) that is accessing the Cryptographic Module. This is only acceptable for FIPS 140-2 level 1 validation.

Role	Type of Authentication	Authentication Data
User	None	N/A
Cryptographic Officer	None	N/A

Table 1: Roles and Required Identification and Authentication

Authentication Mechanism	Strength of Mechanism
None	N/A

Table 2: Strengths of Authentication Mechanisms

5.2 Access Control Policy

The Cryptographic Module supports two roles: User and Cryptographic Officer. An operator performing a service within any role can read/write cryptographic keys and critical security parameters (CSP) only through the invocation of a service by use of the Cryptographic Module API. Thus, that user can read/write the cryptographic keys and CSPs that the given API call allows. The type of services corresponding to each of the supported roles is described in the table below.

Role	Authorized Services
User	• Symmetric Encryption/Decryption – AES – CAST-128 – CAST3 – DES – IDEA – RC2 – RC4 – Rijndael(256-bit block size) – Triple-DES • Asymmetric Encryption/Decryption – RSA • Digital Signature Generation/Verification – DSA – ECDSA – RSA • Hash Generation – MD2 – MD5 – SHA-1 – SHA-256 – SHA-384 – SHA-512

	• MAC Generation – CAST-128 MAC – DES MAC – IDEA MAC – Triple-DES MAC • Keyed-Hash MAC Generation – HMAC-MD5 – HMAC-SHA1 • Key Agreement – Diffie-Hellman – SPEKE • Random Number/Seed Generation – FIPS 186-2 using SHA1 – ANSI X9.31 using DES – ANSI X9.31 using AES256
Cryptographic Officer	▪ Initialization of the Cryptographic Module ▪ Initiate Cryptographic Module Self Tests ▪ Key Input/Output/Generation – AES – CAST-128 – CAST3 – DES – Diffie-Hellman – DSA – ECDSA – HMAC-MD5 – HMAC-SHA1 – IDEA – RC2 – RC4 – Rijndael – RSA – SPEKE – Triple-DES ▪ CSP Input/Output – Output Cryptographic Module Status – Input PKCS11 PIN ▪ All Services of the User role

Table 3: Services Authorized for Roles

The following is a list of the validated FIPS approved services (including appropriate algorithm certificates) that can be used in FIPS mode:

Role	Authorized Services
User	• Symmetric Encryption/Decryption – AES (NIST certificate #193) – DES (NIST certificate #279) – Triple-DES (NIST certificate #289) • Digital Signature Generation/Verification – DSA (NIST certificate #122) – RSA (NIST certificate #30)

	• Hash Generation – SHA-1 (NIST certificate #273) – SHA-256 (NIST certificate #273) – SHA-384 (NIST certificate #273) – SHA-512 (NIST certificate #273) • Keyed-Hash MAC Generation – HMAC-SHA1 (NIST certificate #8) • Random Number/Seed Generation – FIPS 186-2 using SHA1 (NIST certificate #40) – ANSI X9.31 using DES (NIST certificate #40)
Cryptographic Officer	▪ Initialization of the Cryptographic Module ▪ Initiate Cryptographic Module Self Tests ▪ Key Input/Output/Generation – AES – DES – DSA – HMAC-SHA1 – RSA – Triple-DES ▪ CSP Input/Output – Output Cryptographic Module Status – Input PKCS11 PIN ▪ All Services of the User role

Table 4: FIPS-approved Services Authorized for Roles

An operator is explicitly in the User or Cryptographic Officer role based upon the services chosen. If any of the User specific services are called, then the operator is in the User role; otherwise the operator is in the Cryptographic Officer role.

Each service within each role can only access the cryptographic keys and CSPs that the service's API defines. The following cases exist:

- A cryptographic key or CSP is provided to an API as an input parameter; this indicates read/write access to that cryptographic key or CSP.
- A cryptographic key or CSP is returned from an API as a return value; this indicates read access to that cryptographic key or CSP.

Service	Cryptographic Keys and CSPs	Types of Access
Symmetric Encryption/Decryption	Symmetric Key	Read/Write
Asymmetric Encryption/Decryption	Asymmetric Key Pair	Read/Write
Digital Signature Generation/Verification	Asymmetric Key Pair	Read/Write
Hash Generation	None	N/A
MAC Generation	Symmetric Key	Read/Write
HMAC Generation	Symmetric Key	Read/Write
Key Agreement	Asymmetric Key Pair	Read/Write
Random Number Generation	Seed	N/A
Initialization of the Cryptographic Module	None	N/A

Initiation of the Cryptographic Module Self Tests	None	N/A
Key Input/Output	Key	Read/Write
CSP Input/Output	CSP	Read/Write

Table 5: Access Rights within Services

Detailed information on which Cryptographic Module APIs belong to each role can be found in the Cryptographic Module Functional Description[FD]. This document specifies a role for each API call, and the CSPs involved in the call.

5.3 Physical Security Policy

The physical security of the cryptographic module is provided by the PC that it is being used on. For more detailed information on the physical security please refer to [UG], [HMM], and [SM].

5.4 Operational Environment

5.4.1 Assumptions

The following assumptions are made about the operating environment of the cryptographic module:

- Unauthorized reading, writing, or modification of the module's memory space (code and data) by an intruder (human or machine) is not possible; this is prevented by the process memory management of the Operating System.
- Replacement or modification of the legitimate cryptographic module code by an intruder (human or machine) is not feasible
- The module is initialized to the FIPS 140-2 mode of operation

5.4.2 Installation and Initialization

The following steps must be performed to install and initialize the JTK cryptographic module for operating in a FIPS 140-2 compliant manner:

- The operating system must be configured to operate securely and to prevent remote login.
- The operating system must be configured to allow only a single user.
- All the jar files and native libraries shipped with the JTK must be copied to the machine on which the JTK is being used.
- The Java runtime environment must be configured to recognize the JTK jar files either by setting the CLASSPATH environment variable or by using the JTK as an installed extension.
- To operate the JTK in a FIPS 140-2 compliant the cryptographic module must be initialized to operate in OPERATIONAL_FIPS mode; this is done by calling `SecurityEngine.initialize(true)`. This will authenticate the cryptographic module and run the necessary FIPS 140-2 start-up tests

5.4.3 Policy

The following policy must always be followed in order to achieve a FIPS 140-2 mode of operation:

- The cryptographic module must only be used by one human operator at a time, and must not be actively shared among operators at any time. Also, there must be only one instance of the cryptographic module loaded into RAM at any give time on any given machine.

- All keys entered into the cryptographic module must be verified as being legitimate and belonging to the correct entity by software running on the same machine as the cryptographic module.
- Virtual memory that exists on the machine when the cryptographic module runs must be configured to reside on a local, not a networked, drive.
- Output of plaintext private or secret cryptographic keys and CSPs on any physical port must be prohibited by the operator of the cryptographic module.
- The above conditions must be upheld at all times in order to ensure continued system security after initial setup of the validated configuration. If the module is removed from the above environment, it is assumed to not be operational in the validated mode until such time as it has been returned to the above environment and re-initialized by the user to the validated condition.

5.5 Mitigation of Other Attacks Policy

The cryptographic module is not designed to mitigate any specific attacks.

Other Attacks	Mitigation Mechanism	Specific Limitations
None	N/A	N/A

Table 6: Mitigation of Other Attacks