Entrust, Inc.

Cryptographic Module Security Policy

Entrust Authority™ Security Toolkit 7.2 for the Java® Platform

Author: Christopher D. Wood
Date: June 11, 2007
Version: \main\5

Table of Contents

1 Revision History

Authors	Date	Version	Comment
Christopher D. Wood	July 27, 2006	\main\1	First Version
Christopher D. Wood	July 28, 2006	\main\2	Minor improvements
Christopher D. Wood	November 1, 2006	\main\3	Changes requested by DOMUS
Christopher D. Wood	November 5, 2006	\main\4	Noting that Triple-DES MAC is vendor affirmed
Christopher D. Wood	June 11, 2007	\main\5	Changes requested by NIST

Contributors	Topics
DOMUS	Guidance on content

2 References

Author	Title
NIST	[1] FIPS PUB 140-2: Security Requirements For Cryptographic Modules, December 2002
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, May 2006
Entrust	[4] Entrust Authority Security Toolkit 7.2 for the Java Platform - Programmer's Guide, July 2006
Entrust	[CMC] Cryptographic Module Classes for the Entrust Authority Security Toolkit 7.2 for the Java Platform, November 2006
Entrust	[CR] Cryptographic Module Validation Cross-Reference for the Entrust Authority Security Toolkit 7.2 for the Java Platform, November 2006
Entrust	[DD] Cryptographic Module Design Description for the Entrust Authority Security Toolkit 7.2 for the Java Platform, November 2006
Entrust	[FD] Cryptographic Module Functional Description for the Entrust Authority Security Toolkit 7.2 for the Java Platform, November 2006
Dell	[UG] Dell OptiPlex GX620 Systems User's Guide – Mini Tower Computer (English), Dell, (http://support.dell.com/support/edocs/systems/opgx620/en/ug/A02/tindex.htm)
Dell	[QRG] Dell OptiPlex GX620 Quick Reference Guide, Dell, 2005 (http://support.dell.com/support/edocs/systems/opgx620/QRG_AMF/K8502A00.pdf)
Sun	[SM] Sun Fire V210 and V240 Servers Service Manual, Sun Microsystems Inc., 2005 (http://www.sun.com/products-n-solutions/hardware/docs/pdf/819-4207-10.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 the Java® Platform (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 140-2, level 1. The JTK cryptographic module is defined as a multi-chip standalone cryptographic module according to FIPS 140-2.

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. A Dell™ OptiPlex™ GX620 Mini Tower Computer with:
    - Intel® Pentium® D dual-core 3.2 GHz Processor
    - 2x1GB DDR2 RAM DIMMs
    - Disk Drives WDC WD1600JS-75NCB1 149.0Gb Disk Drive
    - HL-DT-ST DVD+-RW GWA4164B CD/DVD Drive
    - Intel(R) 82945G Express Chipset Family 224Mb
    - NEC 1..44MB Floppy Disk Drive
    - Sound Devices SoundMAX Integrated Digital Audio
    - Bradcom NetXtreme 57xx Gigabit Controller

    Operating System:

- Windows XP Professional SP1

Java Runtime Environment:

- Sun JRE 5.0

A detailed technical description of the Dell Optiplex GX620 platform is included in [UG] and [QRG].

2. A Sun Fire V210 Server with:
   - One 1.34 GHz UltraSPARC IIIi processor
   - 2 GB main memory
   - One hot-swap Ultra160SCSI 73 GB disk
   - One Slim-line ATAPI DVD-ROM
   - Four 10/100/1000Base-T Ethernet Ports
   - One TIA/EIA-232-F (RJ45) Port
   - One TIA/EIA-232-F asynchronous (DB9) Port
   - One Ultra160SCSI multimode (SE/LVD)
   - Two OHCI-1.0 Compliant Interfaces, supporting dual speeds of 12 and 1.5 Mbits/sec each
   - One 64 bit 33/66MHz full-length PCI 2.2 compliant slot.
   - 459 watt power supply

   Operating System:
   - Solaris 10

   Java Runtime Environment:
   - Sun JRE 5.0

A detailed technical description of the Sun Fire V210 Server 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.

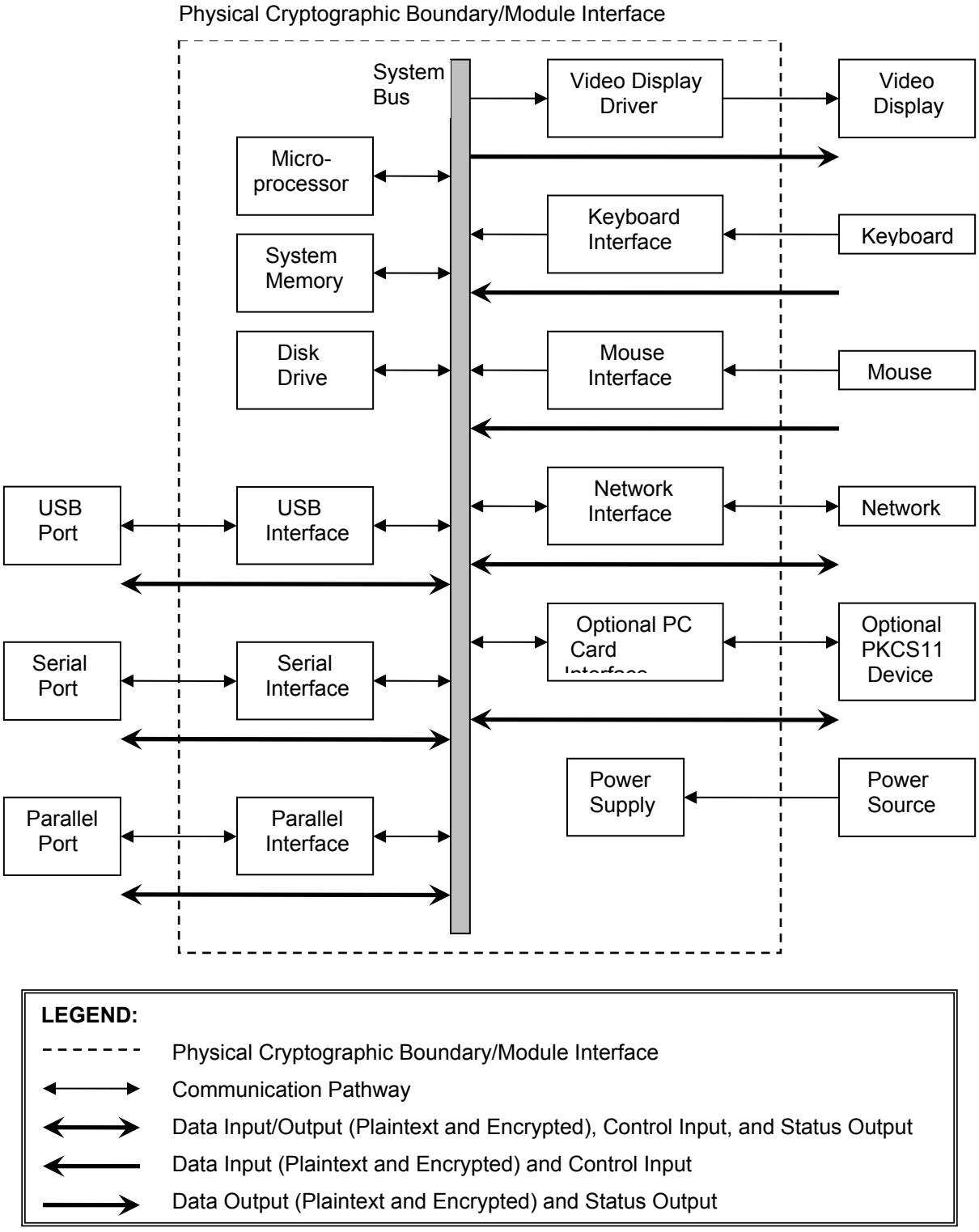

Figure 1: Cryptographic module block diagram for hardware (Physical).

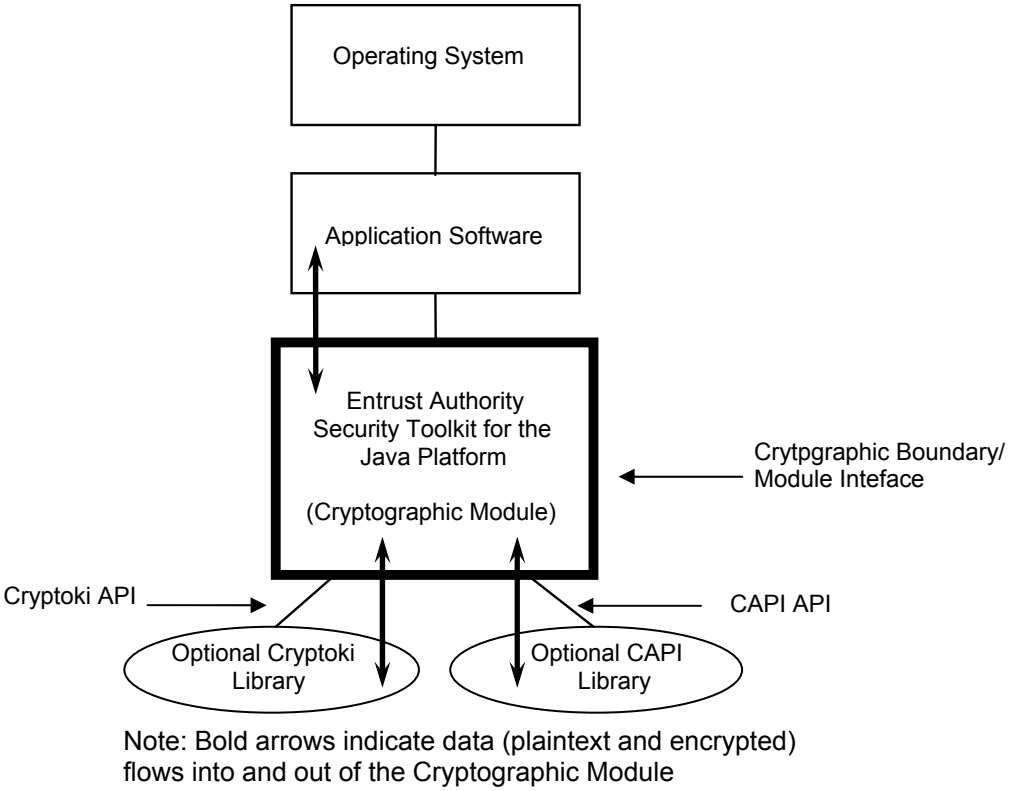

Figure 2: Cryptographic module block diagram for software (Logical).

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
- Key Derivation Algorithms

4.4 Module Ports and Interfaces

The JTK cryptographic module is considered according to the requirements of FIPS 140-2 to be a multi chip standalone module. The table below describes a mapping of logical interfaces to physical ports:

FIPS 140-2 Interface	Logical Interface	Physical Interface
Data Input Interface	Input parameters of module function calls	Ethernet/Network Port, USB Port, Parallel Port
Data Output Interface	Output parameters and return values of module function calls	Ethernet/Network Port, USB Port, Parallel Port
Control Input Interface	Module control function calls	Keyboard and Mouse
Status Output Interface	Return values from module status function calls	Monitor
Power Interface	Initialization function	Power Interface

Table 1: Mapping Logical Interfaces to Physical Ports

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 2: Roles and Required Identification and Authentication

Authentication Mechanism	Strength of Mechanism
None	N/A

Table 3: Strengths of Authentication Mechanisms

5.2 Access Control Policy

The cryptographic module supports two roles: User and Cryptographic Officer. Each service is explicitly assumed by the assigned role. 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	Cryptographic Keys and CSPs	Access Type
Cryptographic Officer	Initialization of the Cryptographic Module	None	Execute
	Initiate Cryptographic Module Self Tests	None	Execute
	Key Input/Output	AES, Triple-DES, RSA, DSA, ECDSA, DH, HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 keys	Execute
	Key Generation (ANSI X9.31 and FIPS 186-2)	AES, Triple-DES, RSA, DSA, ECDSA, DH, HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 keys	Execute, Write
	Module Status	None	Read
User	Symmetric Encryption/Decryption	AES, Triple-DES keys	Execute
	Digital Signature Generation/Verification	DSA, ECDSA, RSA keys	Execute
	Hash Generation	None	Execute
	MAC Generation	AES, Triple-DES, HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 keys	Execute
	Key Agreement	DH keys	Execute
	Asymmetric Key Wrapping	RSA keys	Execute
	Random Number/Seed Generation	None	Execute

Table 4: Services Authorized for Roles

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

FIPS Approved Algorithm	Certificate Number
AES	#443
Triple-DES	#467
DSA	#187
ECDSA	#34
RSA (Signature Generation and Verification)	#168
SHS	#510
HMAC	#209
RNG (ANSI X9.31 and FIPS 186-2)	#231
Triple-DES MAC (vendor affirmed)	#467

Table 5: FIPS-Approved Algorithms

The following is a list of Allowed algorithms (including appropriate key sizes) that can be used in FIPS mode:

- RSA (Key Wrapping: Encryption/Decryption)
  - This key establishment methodology provides between 80 and 256 bits of encryption strength; supported key sizes: [1024, 1032, …, 15360] bits.
- Diffie-Hellman (Key Agreement)
  - This key establishment methodology provides 80 bits of encryption strength; supported key sizes: 1024-bits.

The following is a list of non-FIPS Approved algorithms that are implemented but cannot be used when operating in FIPS mode:

- CAST128
- CAST3
- DES
- IDEA
- RC2
- RC4
- Rijndael
- SPEKE
- ElGamal
- MD2
- MD5
- DES MAC
- IDEA MAC
- CAST128 MAC
- HMAC-MD2
- HMAC-MD5

5.3 Self-Tests

The cryptographic module contains the following self-tests to verify its correct operation; these tests are automatically run during initialization in the FIPS Approved Mode of operation:

Power-On Self-Tests:

- Software integrity test using HMAC-SHA-1
- RNG KAT and continuous test for ANSI X9.31 and FIPS 186-2
- AES KAT for encrypt/decrypt
- Triple-DES KAT for encrypt/decrypt
- SHA-1, SHA-256, SHA-384, SHA-512 KATs
- Triple-DES MAC KAT
- HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 KATs
- AES CMAC KAT
- Triple-DES CMAC KAT
- RSA/SHA1, RSA/SHA256, RSA/SHA384, RSA/SHA512 KATs for sign/verify
- RSA pair-wise consistency test
- DSA pair-wise consistency test
- ECDSA pair-wise consistency test

Conditional Tests:

- RNG continuous test for ANSI X9.31 and FIPS 186-2 RNG
- RSA pair-wise consistency test
- DSA pair-wise consistency test
- ECDSA pair-wise consistency test

5.4 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], [QRG], and [SM].

5.5 Operational Environment

5.5.1 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.

5.5.2 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:

- 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

5.5.3 Policy

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

- 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.
- Input/Output of plaintext private or secret cryptographic keys and CSPs on/from any physical port must be prohibited by the operator of the cryptographic module.
- Cryptographic algorithms are accessed through the Java Cryptography Architecture and must be requested from the Entrust and or IAIK cryptographic service provider. All FIPS approved algorithms must always be requested from the Entrust cryptographic service provider only.
- 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.

5.5.4 Module Operator

FIPS 140-2 states that when using a software cryptographic module, the operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded). NIST has since provided further implementation guidance[3] regarding this matter:

"Software cryptographic modules implemented in client/server architecture are intended to be used on both the client and the server. The cryptographic module will be used to provide cryptographic functions to the client and server applications. When a crypto module is implemented in a server environment, the server application is the user of the cryptographic module. The server application makes the calls to the cryptographic module. Therefore, the server application is the single user of the cryptographic module, even when the server application is serving multiple clients."

This indicates that for an application built on the JTK cryptographic module, the application is always the single user of the cryptographic module even when multiple applications are running concurrently. This permits multiple concurrent JTK based applications to be running on the same machine in a FIPS 140-2 compliant manner.

5.6 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