[redacted]

Entrust, Inc.

Entrust Entelligence™ Kernel-Mode Cryptomodule Security Policy

Date:	25 September 2008
Document version:	\main\9
Cryptomodule version:	1.1

Entrust
Entelligence™

Table of Contents

1 Revision History

Authors	Date	Version	Comment
R. Lockhart	December 17, 2007	\main\1	First version
R. Lockhart	January 04, 2008	\main\2	Minor updates
R. Lockhart	January 24, 2008	\main\3	Added Dell 745 reference, added triple-DES algorithm to Table 5.
R. Lockhart	May 28, 2008	\main\4	- sec. 3.2: updated Vista 64 OS description - sec. 3.2: corrected control arrow at power supply - sec. 3.3: indicated static lib name - sec. 4.2: removed key output - sec. 4.2: added certificate numbers - sec. 4.7: added table of security levels
R. Lockhart	May 29, 2008	\main\5	- sec. 3.2: added processors - table 5: changed wording for triple-DES - sec. 4.7: physical security N/A
R. Lockhart	May 30, 2008	\main\6	Minor corrections.
R. Lockhart	September 11, 2008	\main\7	- title pg: consistent module name - sec 3.3: module definition - sec 4.2: role for zeroization - sec 4.3: added on-demand tests - sec 4.2: AES and T-DES keys accessed during self tests - added section 4.8 on EMI/EMC - added section 4.9 on key management
R. Lockhart	September 16, 2008	\main\8	Improved description of triple-DES key use in Table 7.
R. Lockhart	September 25, 2008	\main\9	- removed version number from doc title. - added “class B” to section 4.8

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
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) [UG] Del OptiPlex GX745 Systems User’s Guide – Mini Tower Computer (English), Dell, (http://support.dell.com/support/edocs/systems/op745/en/UG_en/index.htm)
Dell	[QRG] Dell OptiPlex GX620 Quick Reference Guide, Dell, 2005 (http://support.dell.com/support/edocs/systems/opgx620/QRG_AMF/K8502A00.pdf) [QRG] Dell OptiPlex GX745 Quick Reference Guide , Dell, (http://support.dell.com/support/edocs/systems/op745/multlang/QRG/EN/JH470A00.pdf)

3 Introduction

This document contains a non-proprietary description of the security policy for the Entrust Entelligence™ Kernel-Mode Cryptomodule (EEKMC). It contains a specification of the rules under which the cryptographic module must operate. These security rules were derived from the requirements of FIPS 140-2 [1].

3.1 Purpose of the Security Policy

There are three major reasons for defining the security policy that is 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 in a product, 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.

3.2 Cryptographic Module Definition

This section defines the cryptographic module that is being submitted for validation to FIPS 140-2, level 1. The EEKMC 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. A software component, the Entrust Entelligence Kernel-Mode Cryptomodule that runs on the above platform and operating system. This component is custom designed and written by Entrust in the 'C' computer language and is identical, at the source code level, for all identified hardware platforms and operating systems. An Application Programming Interface (API) is defined as the interface to the cryptographic module.

The EEKMC has three main platform / Operating System configurations as follows:

	Platform	Operating System	Processor
1.	Dell™ OptiPlex™ GX620 Mini Tower Computer	Microsoft Windows XP Professional SP2	Intel Pentium D
2.	Dell™ OptiPlex™ GX620 Mini Tower Computer	Microsoft Windows Vista Enterprise, 32-bit edition	Intel Pentium D
3.	Dell™ OptiPlex™ GX745 Mini Tower Computer	Microsoft Windows Vista Ultimate SP1, 64-bit edition	Intel Core 2 Duo

The cryptographic module is also suitable for platforms from the same or other manufacturers, based on compatible processors with equivalent or greater system resources, and equivalent or later Operating System versions.

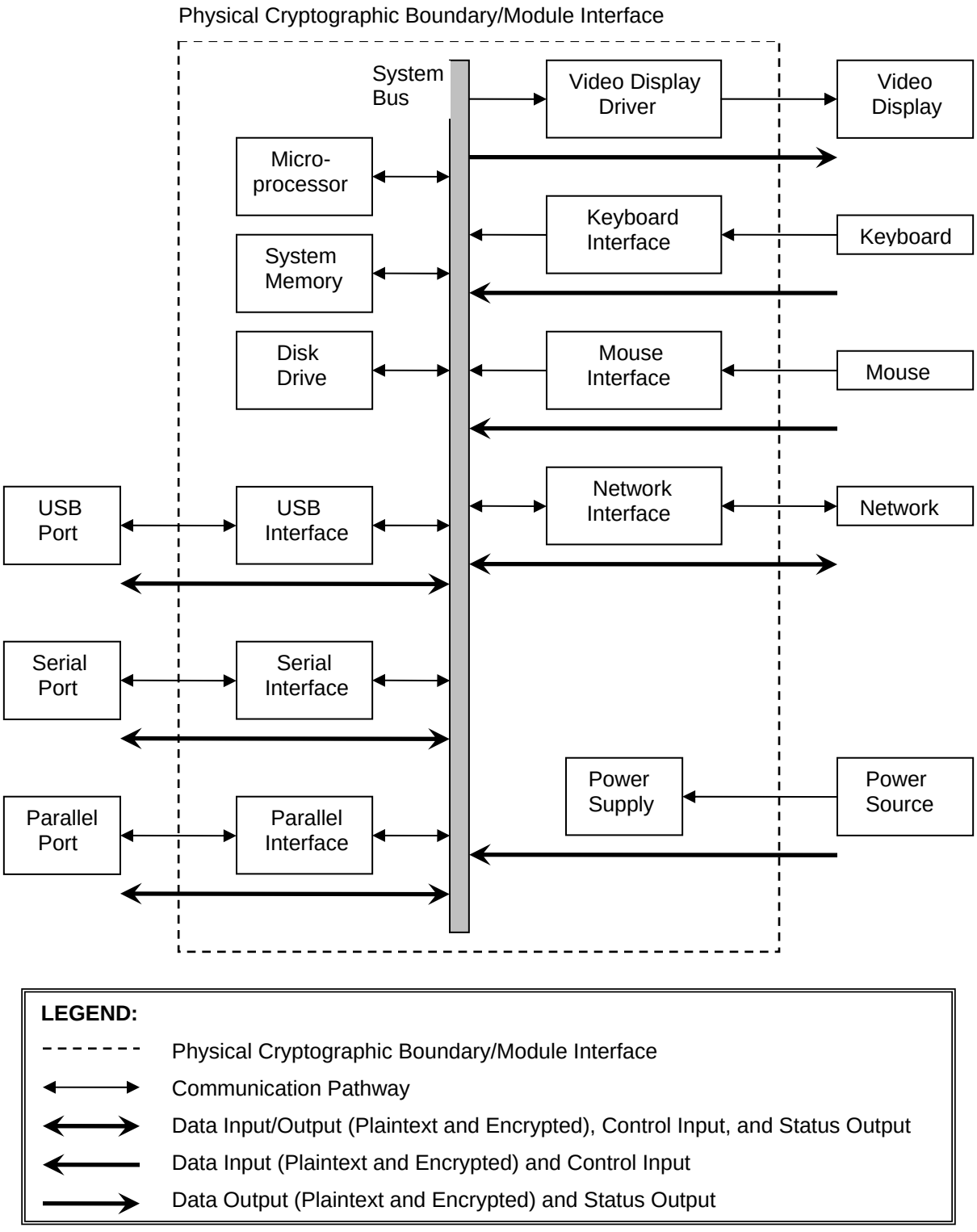
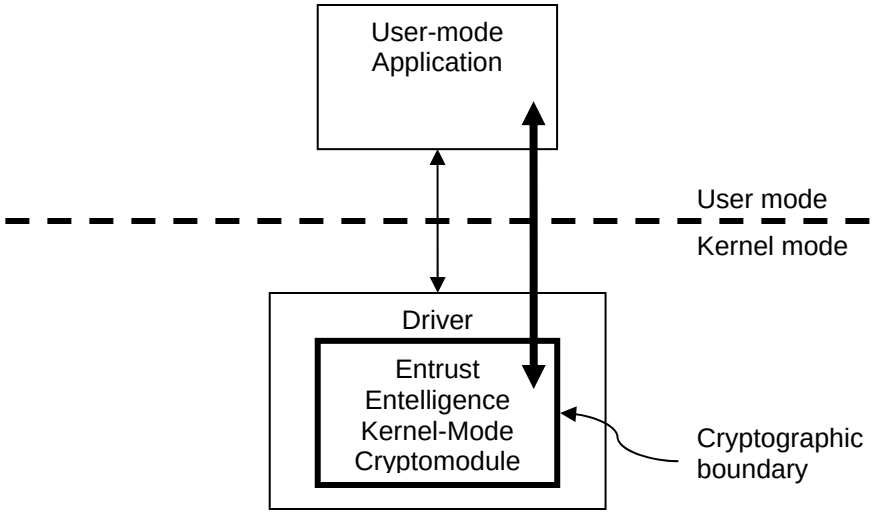


Figure 1: Cryptographic module block diagram for hardware (Physical).



Note: Bold arrows indicate data (plaintext and encrypted) flows into and out of the Cryptographic Module

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

3.3 Cryptographic Module Description

The cryptographic module consists of executable object code which is intended for use in a Windows kernel-mode driver. 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 Entrust Entelligence Kernel-Mode Cryptomodule Functional Specification companion document.

3.4 Module Ports and Interfaces

The 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

4 Specification of the Security Policy

4.1 Identification and Authentication Policy

The cryptographic module neither identifies nor authenticates 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

4.2 Access Control Policy

The cryptographic module supports two roles: User and Cryptographic Officer. The type of services corresponding to each of the supported roles is described in the table below. The roles are implicitly assumed by the operator, based on the services executed.

Role	Authorized Services	Cryptographic Keys and CSPs	Access Type
Cryptographic Officer	Initialization of the Cryptographic Module	None	Execute
	Initiate Cryptographic Module Self Tests	AES and Triple-DES	Execute
	Key Input	AES	Execute
	Module Status	None	Read
	Uninstall module / zeroize keys	AES	Execute
User	Symmetric Encryption/Decryption	AES	Execute

Table 4: Services Authorized for Roles

The following is a list of the validated FIPS Approved algorithms that can be used by the operator of the cryptomodule:

FIPS Approved Algorithm	Certificate Number	Usage
AES-ECB	738	Symmetric encryption/decryption
Triple-DES	655	Software integrity check
Triple-DES MAC	Triple-DES cert. #655, vendor affirmed	Software integrity check

Table 5: FIPS-Approved Algorithms

4.3 Self-Tests

The cryptographic module contains the following self-tests to verify its correct operation The following tests are automatically run during initialization of the module:

Power-On Self-Tests:

- Software integrity test using Triple-DES-MAC
- AES encrypt/decrypt known-answer test
- Triple-DES known-answer test

Conditional Tests:

- none

The following tests can be run on demand by calling the CMRunAlgorithmSelfTests() function:

- AES encrypt/decrypt known-answer test
- Triple-DES known-answer test

The following tests can be run on demand by calling the CMCheckSoftwareIntegrity() function:

- Software integrity test using Triple-DES-MAC

4.4 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] and [QRG].

4.5 Operational Environment

4.5.1 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.

4.5.2 Installation and Initialization

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

- The cryptographic module object code must be embedded in a Windows kernel mode driver executable file, which must then be installed on the computing platform as per the conventions of the Windows Operating System.
- The cryptographic module must be initialized to operate in OPERATIONAL mode; this is done by calling `CMInitialize(executablePath)`. This function will check the software integrity of the cryptographic module and run the necessary FIPS 140-2 start-up tests.
- Each instance of the cryptomodule must be associated with only one application program.
- The operating system must be restricted to a single user at a time. In other words, multiple concurrent operators are not allowed.

4.5.3 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 of plaintext private or secret cryptographic keys and CSPs on any physical port must be prohibited by the operator of the cryptographic module.
- Only the following functions of the cryptographic module may be used when performing cryptographic operations:
    - `AESKeySchedule()`
    - `AESCleanup()`
    - `AESEncryptBlock32()`
    - `AESDecryptBlock32()`

  .

- 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.

4.6 Mitigation of Other Attacks Policy

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

4.7 Security Levels

The cryptographic module is designed to meet the following FIPS 140-2 security levels:

Area	Security Level
Cryptographic module specification	1
Cryptographic module ports and interfaces	1
Roles, services, and authentication	1
Finite state model	1
Physical security	N/A
Operational environment	1
Cryptographic key management	1
EMI/EMC	1
Self-tests	1
Design assurance	1
Mitigation of other attacks	N/A

Table 6: Security Levels

4.8 EMI/EMC

The hardware computing platforms listed in section 3.2 comply with the limits for a Class B digital device pursuant to Part 15 of the Federal Communications Commission (FCC) Rules.

4.9 Key Management

CSP	Key Type	Generation/Input	Output	Storage	Zeroization	Use
AES key	AES	Input by the calling application	Never exits the module	RAM while in use	When the module is uninstalled or by calling AESCleanup( ) function	Encryption/ decryption
Triple-DES key	3-key triple-DES	N/A	Never exits the module	Hardcoded in module binary code	When module is uninstalled	Creates MAC for software integrity check

Table 7: Key Management