Lumension Security, Inc.
Lumension Cryptographic Kernel

(Software Version: 1.0)

FIPS 140-2
Non-Proprietary Security Policy

Level 2 Validation

Document Version 1.2

Prepared for:

Lumension Security, Inc.
15880 N. Greenway Hayden Loop, Suite 100
Scottsdale, AZ 85260
Phone: (888) 725-7828
Fax: (480) 970-6323
www.lumension.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050
Fax: (703) 267-6810
www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for Lumension Security, Inc.'s Lumension Cryptographic Kernel (LCK). This Security Policy describes how the LCK meets the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) requirements for cryptographic modules as specified in Federal Information Processing Standards Publication (FIPS) 140-2. This document also describes how to run the module in its Approved FIPS 140-2 mode of operation. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module.

The LCK is referred to in this document as the cryptographic module, the software module, or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Lumension website (www.lumension.com) contains information on the full line of products from Lumension.
- The NIST Cryptographic Module Validation Program (CMVP) website (csrc.nist.gov/groups/STM/index.html) contains information about the FIPS 140-2 standard and validation program. It also lists contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine document
- Executive Summary document
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Lumension. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 validation documentation is proprietary to Lumension and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Lumension.

2 Lumension Cryptographic Kernel

2.1 Overview

Lumension Security is a leading global security management company that develops, integrates, and markets security software solutions. Lumension's Sanctuary Device Control minimizes the risk of data theft from removable devices through its policy-based enforcement of device use that controls the flow of data to and from network endpoints. Their Sanctuary Application Control product provides policy-based enforcement of application use to secure endpoints from malware, spyware, zero-day threats, and unwanted or unlicensed software.

Lumension markets and sells policy-based device control and application control solutions as separate offerings, but also packages them together to provide a more comprehensive solutions suite. Sanctuary (soon to be re-branded as the Lumension Endpoint Security Suite) provides unified protection of all enterprise endpoints (including laptops, thin clients, and desktops) and control of applications and devices. Figure 1 provides a depiction of Sanctuary in use.

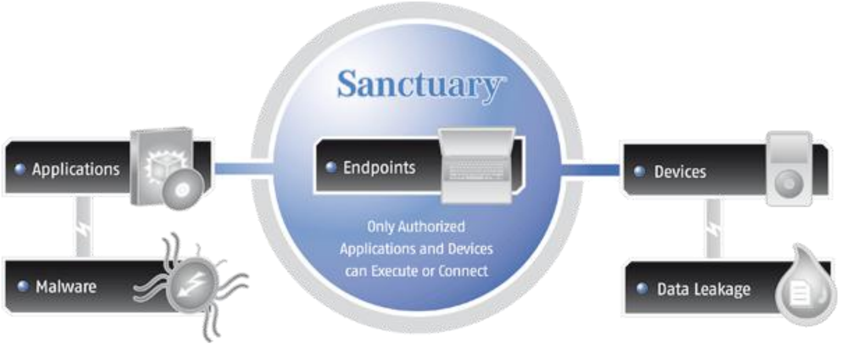

Sanctuary
Applications
Malware
Endpoints
Only Authorized Applications and Devices can Execute or Connect
Devices
Data Leakage

Figure 1 – Example Application/Device Control Scenario

Sanctuary enables administrators to quickly identify devices and applications within a corporate network and establish or enforce device control policies across the network from one or more centralized servers. Policies can be set for individual devices and applications on a network, and permissions can be assigned (by application, device class, specific device, or specific media) to users, user groups, or to a specific computer. By employing a "whitelist" approach, Sanctuary:

- enables only authorized devices to connect to a network or protected endpoint. Unauthorized device access is prohibited by default.
- enables only authorized applications to execute on a network or protected endpoint. Unauthorized applications are prohibited from executing.

Device policies are linked to user and user group information stored in Microsoft Active Directory (AD) or Novell eDirectory and can enforce time constraints, data amounts or transfer limits, and other types of constraints. Additionally, administrators may enforce the encryption of information transferred to removable media while providing centralized encryption key management.

Servers in multi-server deployments communicate with each other over Transmission Control Protocol (TCP) connections, and can optionally be configured to secure these communications using Transport Layer Security (TLS). Endpoints, or clients, are each configured to communicate with only a single server, and they also use TCP (and TLS if so configured).

To improve modularity and maintainability, Lumension has isolated their products' cryptographic functionality from the other functional areas. Lumension has co-located all necessary cryptographic services into one user-mode dynamic link library (DLL) which uses the FIPS 140-2 validated Crypto++ library. This DLL is referred to as the Lumension Cryptographic Kernel (LCK), and it provides cryptographic services to all Sanctuary and Lumension Endpoint Security Suite products.

In FIPS 140-2 terminology, the Lumension Cryptographic Kernel is a multi-chip standalone module that meets the Level 2 FIPS 140-2 requirements. The module was tested and found to be compliant with FIPS 140-2 requirements on the following platforms:

- Dell Optiplex GX620 with an Intel Pentium D CPU running MS Windows Server 2003 Standard, Version 5.2 Service Pack 2 (SP2) (32-bit version)
- Dell Optiplex GX620 with an Intel Pentium D CPU running MS Windows XP Professional, Version 5.1 SP2 (32-bit version)
- Dell PowerEdge 2850 with an Intel Xeon CPU running MS Windows Server 2003 Standard x64, Version 5.2 SP2 (64-bit version)
- Dell PowerEdge 2850 with an Intel Xeon Processor running Windows XP Professional x64, Version 5.2 SP2 (64-bit version)

Note that the DLL name is dependent upon the operating environment for which it was built. For 32-bit environments, the library is named LCK.dll; for 64-bit environments, the library is named LCK64.dll.

The LCK is validated at the following FIPS 140-2 section Levels:

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	2
2	Cryptographic Module Ports and Interfaces	2
3	Roles, Services, and Authentication	2
4	Finite State Model	2
5	Physical Security	N/A[1]
6	Operational Environment	2
7	Cryptographic Key Management	2
8	EMI/EMC[2]	2
9	Self-tests	2
10	Design Assurance	2
11	Mitigation of Other Attacks	N/A

2.2 Cryptographic Boundary

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, there are no physical protection mechanisms implemented; the module must rely on the physical characteristics of the host machine. The physical cryptographic boundary of the LCK is defined by the hard metal enclosure around the computer on which it runs. The module supports the physical interfaces of a Server or a General Purpose Computer (GPC). The physical interfaces include integrated circuits of the motherboard, the central processing unit (CPU), random access memory (RAM), read-only memory (ROM), device

[1] N/A – Not Applicable
[2] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

case, power supply, and fans. Other devices may be attached to the server, such as a display monitor, keyboard, mouse, floppy disk drive, CD-ROM drive, fixed disk drive, printer, audio adapter, or network adapter. See Figure 2 for a standard server block diagram.

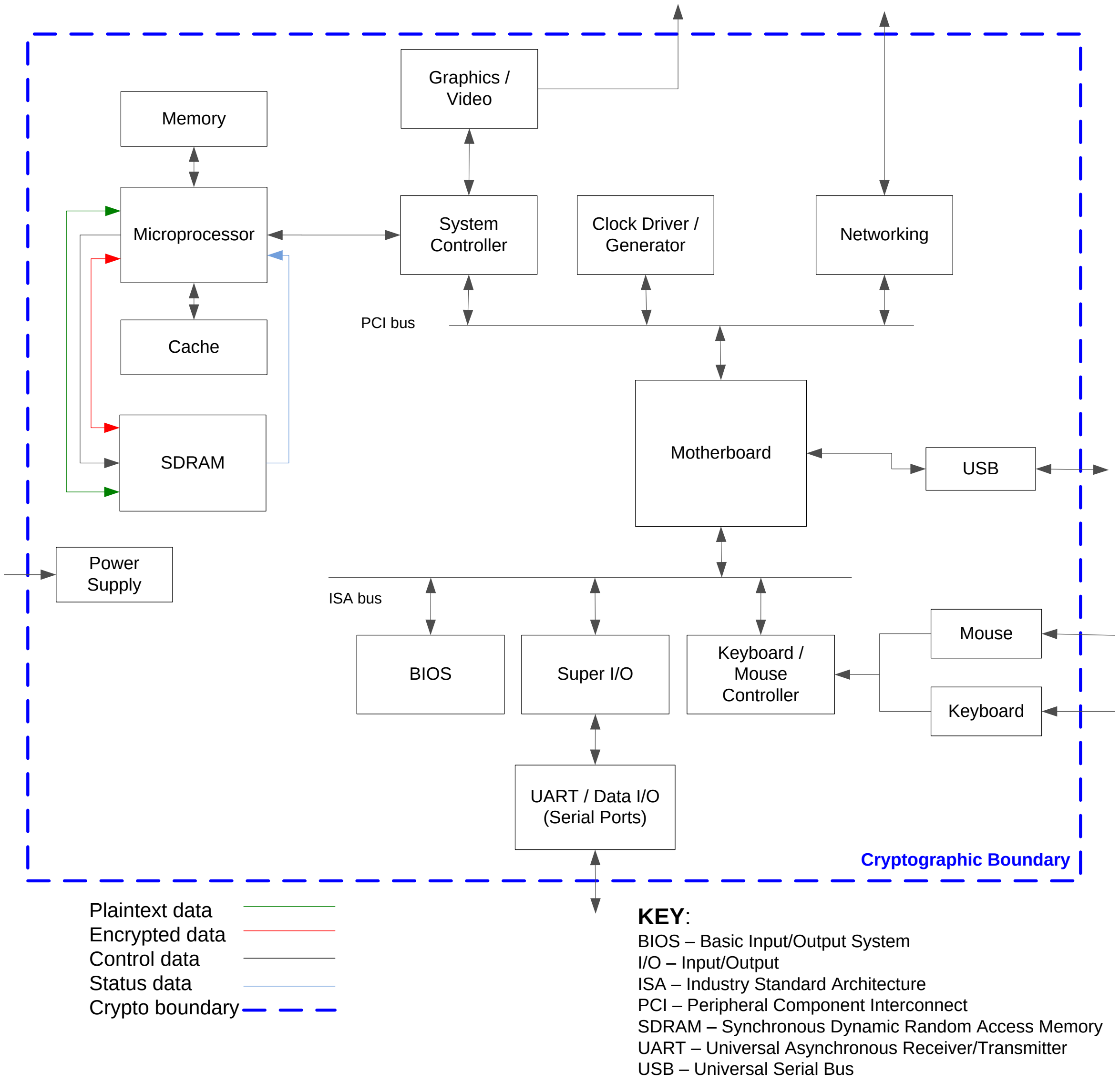

Graphics / Video
Memory
Microprocessor
System Controller
Clock Driver / Generator
Networking
Cache
PCI bus
SDRAM
Motherboard
USB
Power Supply
ISA bus
BIOS
Super I/O
Keyboard / Mouse Controller
Mouse
Keyboard
UART / Data I/O (Serial Ports)
Cryptographic Boundary
Plaintext data
Encrypted data
Control data
Status data
Crypto boundary
KEY:
BIOS – Basic Input/Output System
I/O – Input/Output
ISA – Industry Standard Architecture
PCI – Peripheral Component Interconnect
SDRAM – Synchronous Dynamic Random Access Memory
UART – Universal Asynchronous Receiver/Transmitter
USB – Universal Serial Bus

Figure 2 – Standard Server Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 3 below shows a logical block diagram of the module. The module is a software cryptographic module running on the platforms specified in Section 2.1. The module's software is entirely encapsulated by the logical cryptographic boundary.

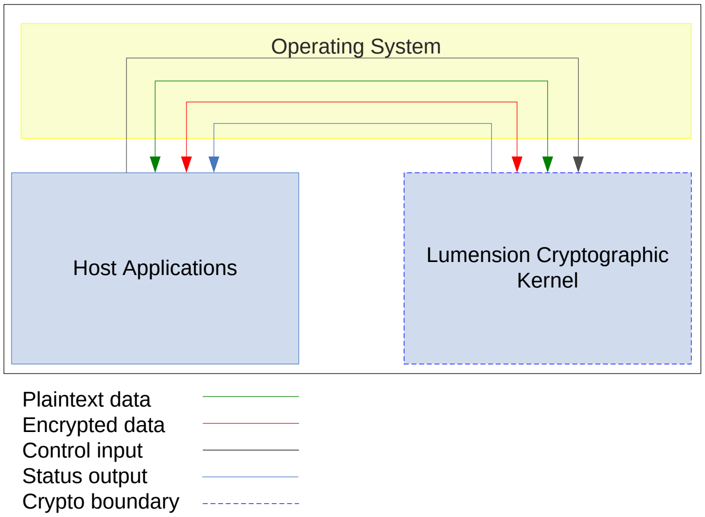


Figure 3 – Logical Block Diagram and Cryptographic Boundary

2.3 Module Interfaces

The module's logical interfaces exist at a low level in the software as an Application Programming Interface (API). The API interface is mapped to the following four logical interfaces:

- Data Input Interface
- Data Output Interface
- Data Control Interface
- Status Output Interface

The module features the physical ports of the host server system, as depicted in Figure 2. The following is a list of typically implemented physical interfaces:

- Keyboard port
- Network ports
- Mouse port
- Display monitor port
- CD-ROM[3] drive
- LED[4] indicators
- Floppy disk
- Power plug/adapter
- Serial ports
- Power switch
- USB ports
- Parallel ports

A mapping of the FIPS 140-2 logical interfaces, the physical interfaces, and the module can be found in Table 2.

[3] CD-ROM – Compact Disc – Read-Only Memory
[4] LED – Light-Emitting Diode

Table 2 – Logical, Physical, and Module Interface Mapping

Logical Interface	Physical Interface Mapping (Standard Server)	Module Mapping
Data Input Interface	Keyboard, mouse, CD-ROM, floppy disk, and serial/USB/parallel/network ports	Function calls that accept, as their arguments, data to be used or processed by the module
Data Output Interface	Floppy disk, monitor, and serial/USB/parallel/network ports	Arguments for a function that specify where the result of the function is stored
Control Input Interface	Keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output Interface	Floppy disk, monitor, LED indicators, and serial/USB/parallel/network ports	Return values for function calls

2.4 Roles, Services, and Authentication

The module supports three operator roles: a Crypto-Officer (CO) role, a User, role, and an Unauthenticated role. The CO role and the User role are used to access symmetric/asymmetric encryption/decryption, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions. The Unauthenticated role has access to services which provide status, compute checksums, and perform encoding/decoding.

Services provided by the module to the CO and User roles, including the Critical Security Parameter (CSP) access required by the services, are listed in Table 3 below. Please note that the keys and CSPs listed in the table indicate the type of access required using the following notation:

- Read: The CSP is read.
- Write: The CSP is established, generated, modified, or zeroized.
- Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

Table 3 – Mapping of Crypto-Officer and User Services to Type of CSP Access

Service	Role		Description	Type of CSP Access
	CO	User		
Installation	✓		Module installation	None
Key Zeroization	✓	✓	Zeroization of keys and CSPs	Symmetric key - Execute Asymmetric key pair - Execute AES key - Execute HMAC key - Execute RSA key pair - Execute PRNG seed - Execute PRNG key - Execute
GenerateRandom	✓	✓	Fill a buffer with random data	Pseudo Random Number Generator (PRNG) seed – Read, execute PRNG key – Read, execute
HashInit	✓	✓	Initialize a hash's state	HMAC key – Read, execute (inclusion of the key depends on algorithm to be used)
HashUpdate	✓	✓	Continue hashing data which was previously started with a call to HashInit	None
HashFinal	✓	✓	Finalize current hash's state	None

Service	Role		Description	Type of CSP Access
	CO	User		
GenerateSymmetricKey	✓	✓	Generate a symmetric key	Symmetric key – Write PRNG seed – Read, execute PRNG key – Read, execute
SymmetricEncrypt	✓	✓	Symmetric encryption of a buffer	Advanced Encryption Standard (AES) key – Read, Execute
SymmetricDecrypt	✓	✓	Symmetric decryption of a buffer	AES key – Read, Execute
GenerateAsymmetricKeys	✓	✓	Generate asymmetric key pair	Asymmetric public/private key pair – Write PRNG seed – Read, execute PRNG key – Read, execute
TransformAsymmetricKey	✓	✓	Tranform asymmetric key in PCKS#8 format	Asymmetric public/private key pair – Read, write
AsymmetricEncrypt	✓	✓	Process data using an asymmetric algorithm	RSA public key – Read, execute
AsymmetricDecrypt	✓	✓	Process data using an asymmetric algorithm	RSA private key – Read, execute
AsymmetricSign	✓	✓	Sign data using an asymmetric algorithm	Asymmetric private key – Read, execute
AsymmetricCheck	✓	✓	Check signature of data using an asymmetric algorithm	Asymmetric public key – Read, execute

The module also provides services for which the operator is not required to assume an authorized role. None of these services modify, disclose, or substitute cryptographic keys and CSPs, or otherwise affect the security of the module. These services are listed in Table 4.

[**NOTE**: These services are also available to the Crypto-Officer and the User.]

Table 4 – Mapping of Unauthenticated Role Services to Type of CSP Access

Service	Description	Type of CSP Access
GetFipsMode	Return the current mode	None
GetFipsState	Retrieve the current FIPS state (used to show status)	None
SelfTest	Run self-tests on demand	None
Encode	Encode binary data in text using the specified alphabet	None
Decode	Decode text data into binary using the specified alphabet	None
GetChecksumSize	Provide the size in bits of the checksum to add to the binary data so it is aligned to the encoded text length	None
ComputeChecksum	Compute the checksum of input data	None

In order to access the cryptographic services offered by the module, operators must first authenticate to authorized roles. The module supports identity-based authentication, and employs the following authentication method (as provided by the operating system) to authenticate the Crypto-Officer and the User.

Table 5 – Authentication Mechanism Employed by the Module

Role	Authentication Type	Authentication Strength
Crypto-Officer or User	Password	The authentication mechanism is provided by the host Operating System. Proper operation of the module requires that the host operating system be configured to enforce a password length of at least 8 (eight) characters long. Alphabetic (uppercase and lowercase) and numeric characters can be used, which gives a total of 62 characters to choose from. With the possibility of repeating characters, the chance of a random attempt falsely succeeding is 1 in 62^8, or 1 in 2 X 10^{14}. Assuming that no password lockout settings were configured, that no delay is configured between password attempts, and that an attacker could attempt 100 password entries per minute, then the probability that a random attempt will succeed is still less than one in 2 X 10^{12} (100 in 2 X 10^{14}). Therefore, the module is sufficiently protected against the random attempt attack for each of the Operating Systems on which it was tested.

2.5 Physical Security

The LCK is purely a software module. As such, it depends on the physical characteristics of the host platform and its protection mechanisms. Thus, physical security requirements do not apply.

2.6 Operational Environment

The module was tested and found to be compliant with FIPS 140-2 requirements on the following platforms:

- Dell Optiplex GX620 with an Intel Pentium D CPU running MS Windows Server 2003 Standard, Version 5.2 Service Pack 2 (SP2) (32-bit version)
- Dell Optiplex GX620 with an Intel Pentium D CPU running MS Windows XP Professional, Version 5.1 SP2 (32-bit version)
- Dell PowerEdge 2850 with an Intel Xeon CPU running MS Windows Server 2003 Standard x64, Version 5.2 SP2 (64-bit version)
- Dell PowerEdge 2850 with an Intel Xeon Processor running Windows XP Professional x64, Version 5.2 SP2 (64-bit version)

The required operating systems on which the module is supported were evaluated to the CC[5] requirements at evaluation assurance level 4+ on 07 February 2008 (Validation Report Number: CCEVS- VR-VID10286-2008). The Common Criteria Evaluation and Validation Scheme Validation Report for the above-referenced operating systems can be found at http://www.niap-ccevs.org/cc-scheme/st/st_vid10184-vr.pdf.

All cryptographic keys and CSPs are under the control of the operating system (OS), which protects the CSPs against unauthorized disclosure, modification, and substitution. The OS uses its native memory management

[5] CC – Common Criteria

mechanisms to ensure that outside processes cannot access the process space used by the module. The module only allows access to CSPs through its well-defined APIs.

Additionally, Lumension affirms that the module also executes in its FIPS-Approved manner (as described in this Security Policy) on other operating systems that are binarily-compatible to those on which the module was tested.

2.7 Cryptographic Key Management

The module uses implementations of FIPS-Approved algorithms listed in Table 6.

Table 6 – FIPS-Approved Algorithms Supported by the Module

Approved Security Function	Certificate Number
Advanced Encryption Standard (AES) 256-bit in CBC[6], ECB[7], OFB[8], CFB[9]128 and CTR[10] modes	1045
Secure Hash Algorithm (SHA) SHA-1, SHA-256, SHA-384, and SHA-512 (byte-oriented)	995
Keyed-Hash Message Authentication Code (HMAC) Using SHA-1, SHA-256, SHA-384, and SHA-512	587
ANSI[11] X9.31 Appendix A.2.4 Pseudo Random Number Generator (PRNG)	596
Elliptic Curve Digital Signature Algorithm (ECDSA) P[12] curves -192, 224, 256, 384, and 521-bit	126
Elliptic Curve Digital Signature Algorithm (ECDSA) K[13] curves -163, 233, 283, 409, and 571-bit	126
Rivest Shamir and Adleman (RSA) Signature generation/verification: 1024, 2048 and 4096 bits	499

The module also implements the following non-Approved algorithms to be used in non-FIPS mode of operation:

- RSA key transport (key wrapping; key establishment methodology provides between 80 and 150 bits of encryption strength)
- MD5[14]
- HMAC-MD5
- Elliptic Curve Integrated Encryption Scheme

The module supports the CSPs listed in Table 7 below.

Table 7 – Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use

[6] CBC – Cipher Block Chaining
[7] ECB – Electronic Codebook
[8] OFB – Output Feedback
[9] CFB – Cipher Feedback
[10] CTR – Counter
[11] ANSI – American National Standards Institute
[12] P Curve – Pseudo-Random Curve
[13] K Curve – Koblitz Curve
[14] MD5 – Message Digest 5

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
Symmetric key	AES ECB, CBC, OFB, CFB128, CTR 256-bit key	Internally generated	Exits in plaintext	Resides in plaintext on volatile memory	On power cycle or API service termination	Used by host application
Asymmetric key pair	RSA 1024-, 2048-, 4096-bit public/private key pair ECDSA P-192, P-224, P-256, P-384, P-512, K-163, K-233, K-283, K-409, K-571 public/private key	Internally generated	Exits in plaintext	Resides in plaintext on volatile memory	On power cycle or API service termination	Used by host application
AES key	AES ECB, CBC, OFB, CFB128, CTR 256-bit key	Externally generated, enters the module in plaintext	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Encrypts/decrypts data
HMAC key	HMAC SHA-1, HMAC SHA-256, HMAC SHA-384, and HMAC SHA-512 key	Externally generated, enters the module in plaintext	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Generates MAC value
RSA public key	RSA 1024-, 2048-, 4096-bit public key	Externally generated, enters the module in plaintext	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Protects a symmetric key during key transport
RSA private key	RSA 1024-, 2048-, 4096-bit private key	Externally generated, enters the module in plaintext	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Retrieves a symmetric key during in key transport
PRNG seed	8-bytes of seed value	Internally generated	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Generates FIPS approved random number
PRNG key	24-bytes of TDES key	Internally generated	Never exits the module	Resides in plaintext on volatile memory	On power cycle or API service termination	Generates FIPS approved random number

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 PRNG implementation to generate cryptographic keys. This PRNG is FIPS-Approved as shown in Annex C to FIPS PUB 140-2.

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output; however keys can be passed to the module as parameters from the application via the APIs. The application using the module is responsible for ensuring that the input or output of secret and private keys is accomplished in encrypted form.

2.7.3 CSP Storage and Zeroization

The module does not persistently store any CSPs. All of the keys and CSPs in Table 7 above reside only on the volatile memory in plaintext and can be zeroized via power cycle. Termination of API services can also zeroize these ephemeral keys and CSPs.

2.8 EMI / EMC

The module is a software module, and depends on the host server systems for its physical characteristics. However the host server platforms have been tested for, and meet, applicable Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. All servers sold in the United States must meet the applicable FCC requirements.

2.9 Self-Tests

The Lumension Cryptographic Kernel automatically performs the following self-tests at power-up:

- Software integrity check using HMAC-SHA-1
- AES Known Answer Test (KAT)
- HMAC SHA-1, HMAC SHA-256, HMAC SHA-384 and HMAC SHA-512 KATs
- SHA-1 and SHA-2s (except SHA-224) KAT
- ANSI x9.31 Appendix A.2.4 PRNG KAT
- RSA pairwise consistency test for sign/verify
- RSA pairwise consistency test for encrypt/decrypt
- ECDSA pairwise consistency test for sign/verify

The Lumension Cryptographic Kernel implements a critical function at power-up that converts an RSA key from PKCS[15] #1 v1.5 format to PKCS #8 format. The module also performs the following conditional self-tests:

- Continuous Random Number Generator Test (CRNGT) for the ANSI X9.31 PRNG
- RSA pairwise consistency test for sign/verify and encrypt/decrypt
- ECDSA pairwise consistency test for sign/verify

The module enters an error state when a power-up self-test fails. Power-up self-tests include the software integrity test, RSA and ECDSA pairwise consistency tests, and KATs for HMAC SHA-1, HMAC SHA-256, HMAC SHA-384, HMAC-SHA-512, PRNG, and AES. The power-up self-tests run through to completion without interruption, and provide no mechanism for data output.

Upon power-up self-test failure, the module will enter a critical error state and will fail to launch in FIPS mode of operation. No data output or cryptographic operations are possible when the module enters the critical error state.

Failure of the RSA or ECDSA pairwise consistency check or CRNGT takes the module into the soft error state. No data output or cryptographic operations are possible when the module enters the soft error state.

[15] PKCS – Public-Key Cryptography Standards

2.10 Design Assurance

Lumension uses Team Foundation Server (TFS) as the configuration management system. Team Foundation Server provides a source control repository, called Team Foundation Version Control (TFVC). TFVC features include check-ins for a group of items or for single changes, branching and merging, shelving, check-in policies, a graphical user interface and a command line interface.

Additionally, Microsoft Visual SourceSafe (VSS) version 6.0 was used to provide configuration management for the module’s FIPS documentation.

2.11 Mitigation of Other Attacks

The module does not claim to mitigate any additional attacks in an approved FIPS mode of operation.

3 Secure Operation

The LCK meets the Level 2 requirements for FIPS 140-2. The sections below describe how to ensure that the module is operating securely.

3.1 Initial Setup

The LCK is distributed via secure download over a TLS channel from a web-based Customer Portal. Login credentials are required for access to the Customer Portal. If proper credentials are given, then the module can be downloaded.

The module is a DLL that will be loaded by a host application. The module's mode of operation is determined by the DWORD registry value at *HKLM\SOFTWARE\Lumension Security\LCK_FIPS_MODE*. When the application loads the module, the module's main entry point checks this value. If the value is "1", then the module will set an internal variable that will allow the host application to access only FIPS-Approved cryptographic services; for all other values, the module will allow the host application to access both FIPS-Approved and non-FIPS-Approved services.

When the host application is first installed, the FIPS-mode registry value is automatically set to "1". This value can be later changed by the CO (see section 3.3.3).

3.2 Power-Up Self-Tests

When the module is powered up, it runs the power-on self-tests automatically; this does not require any action on the part of the CO. If the power-up self-tests are passed, the module is deemed to be operating in FIPS mode.

3.3 Crypto-Officer Guidance

3.3.1 Initialization

The Crypto-Officer is an authorized IT administrator responsible for installing and initializing the module. The module is installed during the process of installing the host application. The CO must ensure that:

- the host application is installed on one of the CC-evaluated OSs listed in Section 2.6;
- the operating system is restricted to a single operator mode of operation;
- the operating system is configured as specified in the CC Validation Report referenced in Section 2.6; and
- the host application is installed on a machine that meets the minimum hardware and software requirements.

3.3.2 Protection of Secret and Private Keys

It is the responsibility of the Crypto-Officer to ensure that any application using the LCK module protects secret and private keys being input or output by doing so in encrypted form.

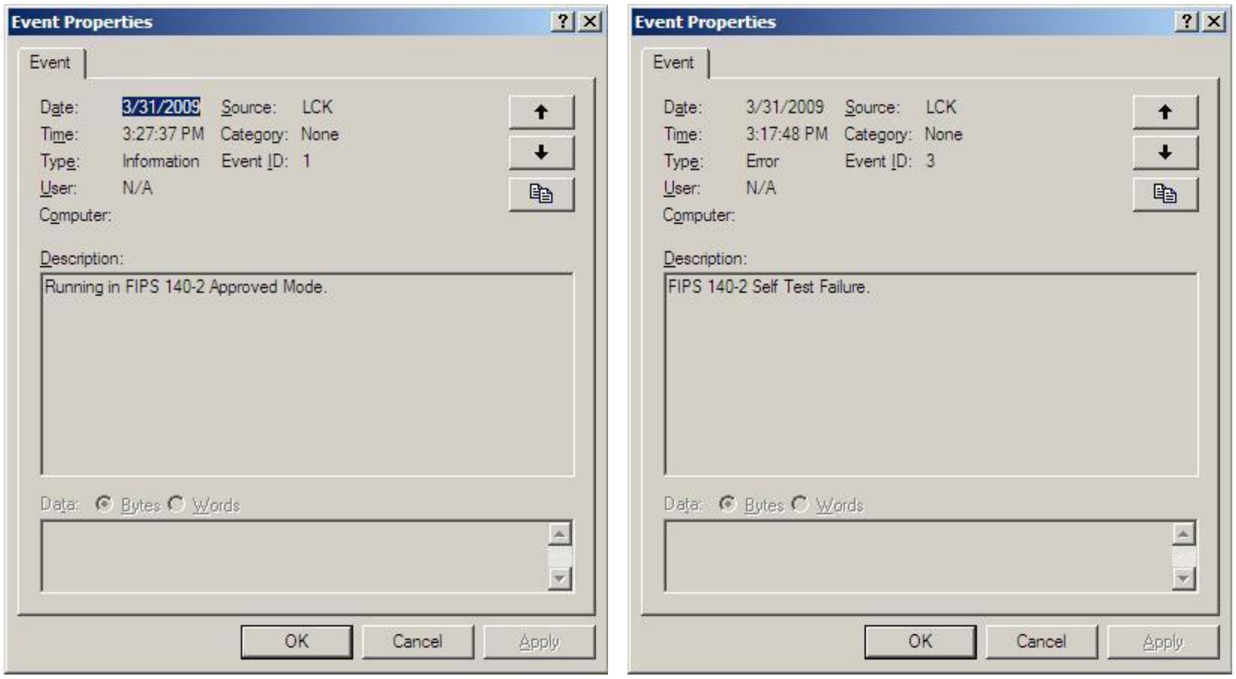
3.3.3 FIPS Mode Registry Value

The FIPS mode registry value is set automatically when its host application is installed. However, it is the CO's responsibility to ensure that the registry value is properly set for the desired mode of operation. The CO must use the host application's management interface to change the registry value, and this action requires that the host application be licensed for FIPS use.

Because the module checks the registry value only at startup, changes to the registry value will have no effect on the module once it is up and operational. To force the module to recognize registry value changes, the CO must perform a reboot or power-cycle to unload and reload the module.

3.3.4 Self-Test Status Monitoring

The CO can view the module's power-up self-test status by checking in the Windows Event log (see Figure 4). When the power-up tests are completed, an indication of overall success or failure is output via the status output interface to the host application.



Date: 3/31/2009 Source: LCK
Time: 3:27:37 PM Category: None
Type: Information Event ID: 1
User: N/A
Computer:
Description:
Running in FIPS 140-2 Approved Mode.
Data: ● Bytes ○ Words

Date: 3/31/2009 Source: LCK
Time: 3:17:48 PM Category: None
Type: Error Event ID: 3
User: N/A
Computer:
Description:
FIPS 140-2 Self Test Failure.
Data: ● Bytes ○ Words

Figure 4 – Module Event Log Display

3.4 User Guidance

The cryptographic functionality of the module (i.e. the collection of User role services) is listed in Table 3 above.

4 Acronyms and Abbreviations

Table 8 – Acronyms and Abbreviations

Acronym/ Abbreviation	Definition
AD	Active Directory
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
BIOS	Basic Input/Output System
CBC	Cipher-Block Chaining
CD-ROM	Compact Disc – Read-Only Memory
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CO	Crypto-Officer
CRNGT	Continuous Random Number Generator Test
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CTR	Counter
DLL	Dynamic-Link Library
ECB	Electronic Codebook
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
HMAC	(Keyed-) Hash Message Authentication Code
I/O	Input/Output
ISA	Industry Standard Architecture
K Curve	Koblitz Curve
KAT	Known Answer Test
LCK	Lumension Cryptographic Kernel
LED	Light Emitting Diode
MAC	Message Authentication Code
MD5	Message Digest 5
N/A	Not Applicable
NIST	National Institute of Standards and Technology

Acronym/ Abbreviation	Definition
OFB	Output Feedback
OS	Operating System
P Curve	Pseudo-Random Curve
PCI	Peripheral Component Interconnect
PKCS	Public Key Cryptography Standard
PRNG	Pseudo Random Number Generator
RSA	Rivest Shamir and Adleman
SDRAM	Synchronous Dynamic Random Access Memory
SHA	Secure Hash Algorithm
SP2	Service Pack 2
TCP	Transmission Control Protocol
TFS	Team Foundation Server
TFVC	Team Foundation Version Control
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VSS	Visual SourceSafe
X64	64-bit Instruction Set
X86	32-bit Instruction Set