VMware Inc.
ACE Encryption Engine

(Software Version: 1.0)

FIPS 140-2
Non-Proprietary Security Policy

Level 1 Validation

Document Version 0.4

Prepared for:

VMware Inc.
3145 Porter Drive
Palo Alto, CA 94304
Phone: (650) 475-5000
Fax: (650) 475 5005
http://www.VMware.com

Prepared by:

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

Revision History

Version	Modification Date	Modified By	Description of Changes
0.1	2007-01-19	Rumman Mahmud	Initial draft.
0.2	2007-04-03	Rumman Mahmud	Modified “Cryptographic Key Management” section.
0.3	2007-04-19	Rumman Mahmud	Inserted Algorithm Certificate numbers
0.4	2007-09-04	Darryl Johnson	Updated to address CMVP comments.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

This document is a non-proprietary Cryptographic Module Security Policy for the ACE Encryption Engine from VMware Inc. This Security Policy describes how the ACE Encryption Engine meets the security requirements of FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

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 National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/cryptval/

The ACE Encryption Engine is referred to in this document as the cryptographic 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 VMware website (www.VMware.com) contains information on the full line of products from VMware.
- The CMVP website (http://csrc.nist.gov/cryptval/) contains 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
- Other supporting documentation as additional references

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

2 ACE Encryption Engine

2.1 Overview

VMware Inc. was founded in 1998 to bring virtual machine technology to industry-standard computers. VMware delivered its first product in 1999. Today, more than 4 million users and 20,000 corporate customers of all types and sizes use VMware software, including 99 of the Fortune 100 companies. The ACE Encryption Engine was created by VMware Inc. in 2004 to extend the capabilities of virtual infrastructure technology to the enterprise desktop.

The ACE Encryption Engine allows virtual machines to be encapsulated into files which can be saved, copied, and provisioned into a virtual machine. The virtual machine allows for fully configured applications, operating systems, BIOS, and virtual hardware, which can be moved from one physical computer to another without needing downtime or maintainance.

VMware's ACE 2.0 is a software solution that delivers enhanced management, security, and usability to standard desktop virtualization products. Using ACE 2.0, an organization can rapidly provision a standardized, secure general purpose computer (GPC) environment — an ACE — to any device in the extended enterprise, regardless of whether it is managed by the ACE administrator. An ACE is a policy-protected virtual machine containing an operating system, applications, and data. Through virtual rights management technology, ACE 2.0 enables desktop administrators to control ACE lifecycles, protect data, and ensure compliance with IT policies including software lifecycle management and access to data and applications.

Unlike other products, ACE 2.0 is a hardware-independent solution that can be provisioned to any GPC and works either connected or disconnected from the enterprise network.

ACE 2.0 is used across an organization to

- Ensure secure, controlled access to enterprise resources from a standardized PC environment called an ACE
- Provide a simplified end-user interface designed specifically for nontechnical Users
- Provide policy-based controls including access, network, and device rights

VMware's ACE allows security administrators to package an entire system with an operating system, applications, and policies, and deploy it to an unmanaged GPC. Once VMware ACE is installed, it offers complete control of the hardware configuration and networking capabilities of the GPC. The VMware ACE application itself is a virtual machine that runs on top of a physical hardware such as a GPC, and simulates the operation of the virtualized system it mimicks.

Figure 1 – Deploying VMware

The VMware ACE Encryption Engine is the kernel of the VMware ACE 2.0 application, and it runs on a GPC. The ACE Encryption Engine contains all the cryptographic software files that enable the VMware ACE application to

perform its cryptographic functions such as hashing, encryption, digital signing, etc. The ACE Encryption Engine is composed of three dynamic-link library (DLL) files: vmcryptolib.dll, ssleay32.dll, and libeay32.dll. The ACE Encryption Engine's logical boundary is shown below in Figure 2.

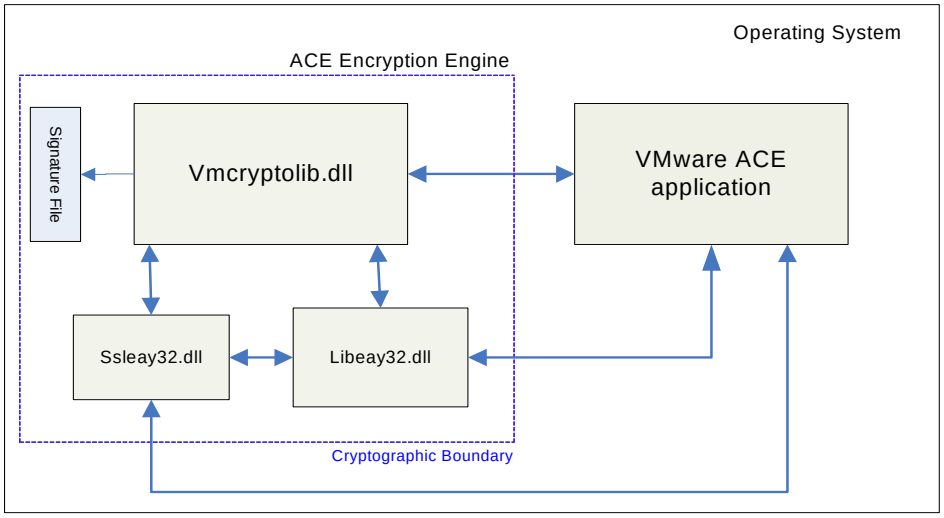

Figure 2 – Logical Boundary of the ACE Encryption Engine

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

2.2 Module Interfaces

The ACE Encryption Engine is a multi-chip stand-alone module that meets overall level 1 FIPS 140-2 requirements. The logical cryptographic boundary of the ACE Encryption Engine is defined by the blue broken line that surrounds the vmcryptolib.dll, libeay32.dll and ssleay32.dll files in Figure 2 **Error! Reference source not found.**.

The module is evaluated for use on a GPC. In addition to the binaries, the physical device consists of the integrated circuits of the motherboard, the central processing unit (CPU), random access memory (RAM), read-only memory (ROM), computer case, keyboard, mouse, video interfaces, expansion cards, and other hardware components included in the computer such as hard disk, floppy disk, CD-ROM drive, power supply, and fans. The physical cryptographic boundary of the module is the hard opaque metal and plastic enclosure of the computer . The block diagram for a standard personal computer (PC) is shown below in Figure 3.

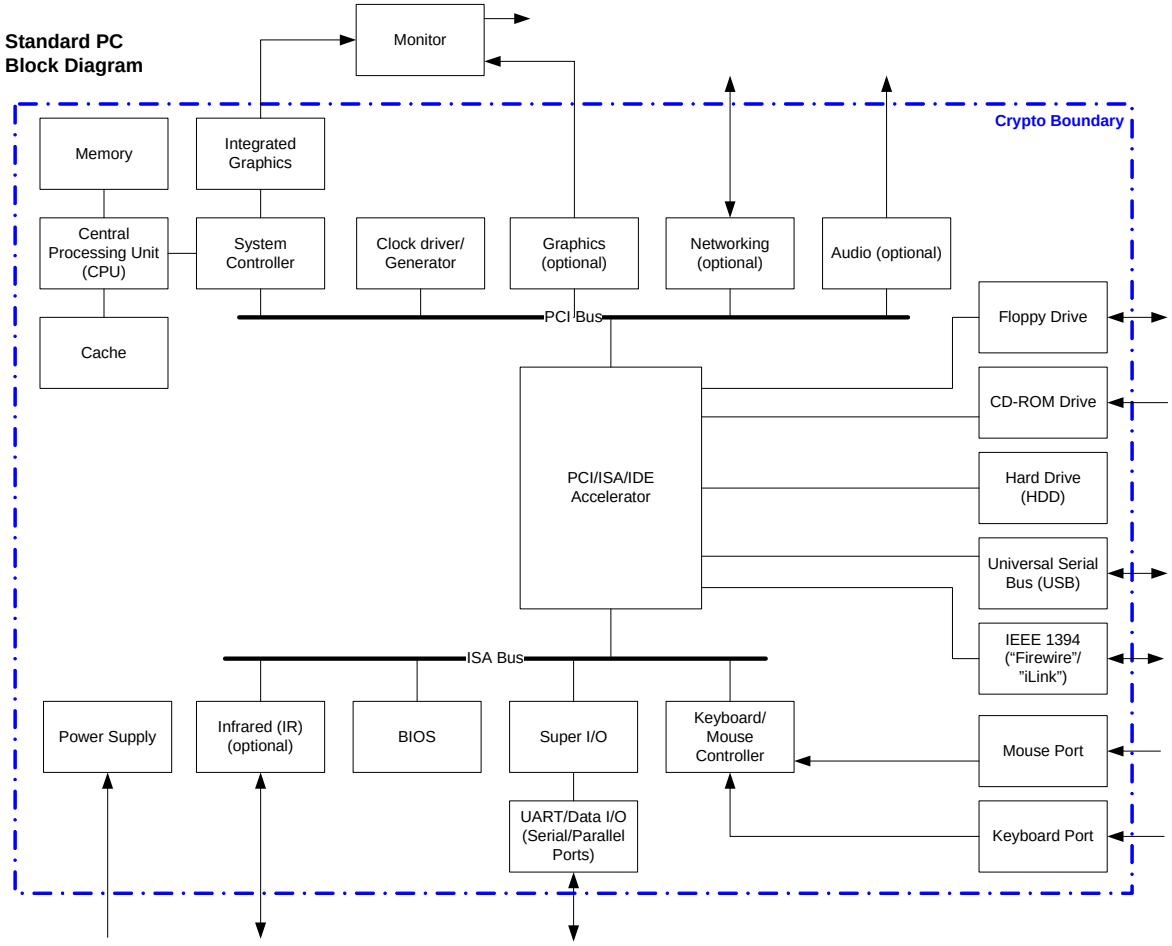

Figure 3 – Standard PC Physical Block Diagram

All of these physical interfaces are separated into logical interfaces defined by FIPS 140-2. Each of these locial interfaces are mapped to the corresponding physical interfaces in Table 2.

Table 2 – Mapping of Logical to Physical Interfaces

Logical Interface	Physical Interface Mapping (Standard PC)	Module Mapping
Data Input Interface	Keyboard, mouse, CD-ROM, floppy drive, and serial/USB/parallel/network ports	Arguments for a function that specify the data to be operated upon by that function.
Data Output Interface	Floppy drive, 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 drive, mouse, and	Function calls utilized to initialize the module and the function calls used to control the

Logical Interface	Physical Interface Mapping (Standard PC)	Module Mapping
	serial/USB/parallel/network port	operation of the module.
Status Output Interface	Floppy drive, monitor, and serial/USB/parallel/network ports	Return values for function calls
Power Interface	Power Switch	Not Applicable

2.3 Roles and Services

There are two roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto Officer (CO) role and a User role. The operator of the module assumes either of the roles based on the operations performed without any authentication. Both of the roles and their responsibilities are described below.

2.3.1 Crypto Officer Role

The Crypto Officer role has the ability to install, uninstall and manage the the ACE Encryption Engine module. The Crypto Officer monitors the ACE Encryption Engine by tracking the event log of the PC's operating system. Descriptions of the services available to the Crypto Officer role are provided in the table below.

Table 3 – Mapping of Crypto Officer Role's Services to Inputs, Outputs, CSPs, and Type of Access

Service	Description	Input	Output	CSP and Type of Access
Installation	Installing the ACE encryption engine module	Command	Module installed	None
Uninstallation	Uninstalling the ACE encryption engine module	Command	Module uninstalled	None
Management	Monitoring, managing and troubleshooting the event log of the operating system	Command	Status output	None

2.3.2 User Role

The User role has the ability to utilize the API's available from the ACE Encryption Engine cryptographic services such as hashing, message authentication, encryption, pseudo-random number generation and disk sector encryption. Also, the module provides TLS services to Users. Descriptions of the services available to the User role are provided in the table below.

Table 4 – Mapping of User Role's Services to Inputs, Outputs, CSPs, and Type of Access

Service	Description	Input	Output	CSP and Type of Access
Hashing	Hashing operation	API call with input parameters	Status output, data hashed	None
Message authentication	Message authentication services	API call with input parameters	Status output, message authentication code (MAC) created	HMAC key – Read
Cipher operation	Cipher operation	API call with input parameters	Status output, data encrypted or decrypted	Symmetric key – Read/Write

Service	Description	Input	Output	CSP and Type of Access
Digital Signing	Digital sign and verify operation	API call with input parameters	Status output, Signature	RSA key pair – Read
Data conversion	Convert entered data into another structure	API call with input parameters	Status output, data reformatted	None
Pseudo-random Number Generation	Generates random numbers	API call with input parameters	Status output, random number generated	Seed – Read
TLS services	Establish TLS channel	API call with input parameters	TLS connection	RSA key pair – Read/Write Symmetric key – Read/Write
Disk Sector Encryption	Encryption of disk sector	API call with input parameters	Status output, disk sector encrypted	AES disk sector encryption key – Read

2.4 Physical Security

The ACE Encryption Engine is a multi-chip standalone module, which is purely a software module and thus physical security requirements do not apply.

2.5 Operational Environment

The module was tested for FIPS 140-2 validation on Windows XP Professional with SP2 and Windows Vista Ultimate operating systems (OS). VMware affirms that the module is binarily compatible for the general purpose Windows 2000 OS and later editions. The OS must be configured for single user mode, per NIST CMVP guidance for FIPS 140-2 compliance. Single user mode configuration instructions for the OS can be found in the "Secure Operation" section of this document.

2.6 Cryptographic Key Management

The ACE Encryption Engine implements the following FIPS-approved algorithms:

In Libeay.dll:

- PRNG Appendix A.2.4 of ANSI X9.31 using 2-key Triple-DES – (certificate #306)
- AES encrypt/decrypt (ECB, CBC mode 128/256 bit) – (certificate #533)
- DSA signature generation and verification (1024 bit) – (certificate #220)
- HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512 (certificate #280)
- RSA sign/verify (1024, 2048, and 4096 bits)– (certificate #241)
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (certificate #603)
- TDES encrypt/decrypt (ECB, CBC mode KO 1,2,3) – (certificate #536)

In VMCryptolib.dll:

- PRNG – FIPS 186-2 General Purpose – (certificate #307)
- AES encrypt/decrypt (ECB, CBC, CTR mode 128/256 bit) – (certificate #534)
- HMAC SHA-1, HMAC SHA-256 (certificate #281)
- SHA-1, SHA-256 (certificate #604)

Additionally, the module utilizes the following non-FIPS approved algorithm implementation:

In Libeay.dll:

- Non-approved FIPS RNG used for seeding the FIPS approved deterministic RNG
- RSA (encrypt/decrypt 1024, 2048, and 4096 bits) (key transport methodology provides between 80-bits and 150-bits of encryption strength)
- DH (Key agreement 1024 bits) (key agreement methodology provides between 80-bits and 150-bits of encryption strength)
- RSA (sign/verify 512 bits)
- MD5

In VMCryptolib.dll:

- Non-approved FIPS RNG used for seeding the FIPS approved deterministic RNG

All secret keys and Critical Security Parameters (CSP) are protected against unauthorized disclosure, modification, and substitution. The module only allows access to CSPs through the module's well-defined commands in the API. The module supports the following cryptographic keys and critical security parameters:

Table 5 – List of Cryptographic Keys, Key Components, and CSPs Present in Libeay.dll

Key	Key Type	Generation/Input	Output	Storage	Zeroization	Use
RSA public key	1024, 2048, and 4096 bit public key	Generated internally or enters the module in plaintext	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used for encryption (key transport) and signature verification
RSA private key	1024, 2048, and 4096 bits private key	Generated internally or enters the module in plaintext	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used for decryption (key transport) and signing
AES Symmetric key	128 and 256 bits ECB, CBC key	Generated internally or enters the module in plaintext or in encrypted form	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used to perform cipher operation on data for TLS channel
TDES Symmetric key	TDES ECB, CBC 1, 2, or 3 key	Generated internally or enters the module in encrypted form	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used to perform cipher operation on data for TLS channel
HMAC key	HMAC key	Generated internally or enters the module in plaintext	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used for message authentication and integrity for TLS channel
PRNG seed	8 bytes of seed value	Generated internally	Never exits the module	Held in volatile memory only in plaintext	Erased when session is closed or on reset	Used to generate pseudo-random number
PRNG seed key	16 bytes of key	Generated internally	Never exits the module	Held in volatile memory only in plaintext	Erased when session is closed or on reset	Used to generate pseudo-random number

Key	Key Type	Generation/Input	Output	Storage	Zeroization	Use
DSA public key	1024 bit public key	Hardcoded in source code	Never exits the module	Stored in non-volatile memory in plaintext	Erased when module is uninstalled	Used for power-up software integrity test

The module generates an HMAC keys internally or enters in plaintext during Transport Layer Security (TLS) handshaking; an Advanced Encryption Standard (AES) or Triple Data Encryption Standard (TDES) Symmetric key is generated internally using FIPS approved PRNG (ANSI X9.31 Appendix A.2.4) during TLS handshaking. Both the ephemeral keys are zeroized when the session is closed or the module is reset/reloaded. The Pseudorandom Number Generator (PRNG) seed and seed key are generated internally and are zeroized when the session is closed or the module is reset or reloaded. The Rivest Shamir and Adleman (RSA) public/private key is generated when the caller provides the certificate containing the key and is zeroized when the TLS context is destroyed. The DSA public key is hardcoded in the source code and is zeroized when the module is uninstalled.

Table 6 – List of Cryptographic Keys, Key Components, and CSPs Present in VMCryptolib.dll

Key	Key Type	Generation/Input	Output	Storage	Zeroization	Use
AES Disk Sector Encryption key	128/256 bits ECB, CBC, CTR key	Generated internally or enters the module in plaintext	Exits the module in plaintext	Held in non-volatile memory in plaintext.	Erased when session is closed or on reset	Used to perform cipher operation on data
HMAC key	HMAC key	Generated internally or enters the module in plaintext	Exits the module in plaintext	Held in volatile memory in plaintext.	Erased when session is closed or on reset	Used for message authentication and integrity validation
PRNG seed	20 bytes of seed value	Generated internally	Never exits the module	Held in volatile memory only in plaintext	Erased when session is closed or on reset	Used to generate pseudo-random number

HMAC keys enter into the module in plaintext. These keys are used for message authentication and integrity checking. VMCryptolib.dll generates an AES Symmetric key internally using FIPS 186-2 Appendix 3.1 PRNG in order to encrypt information stored on the workstation. The AES Disk Sector Encryption key resides only in volatile memory and is zeroized after session is over or on reset. The Pseudorandom Number Generator (PRNG) seed is generated internally and is zeroized when the session is closed or the module is reset or reloaded.

2.7 Self-Tests

The ACE Encryption Engine performs the following self-tests at power-up:

For VMCryptolib.dll:

- Software integrity check (using DSA)
- Known Answer Tests (KATs)
  - AES KAT (ECB, CBC, CTR modes 128/256 bit)
  - HMAC SHA-1 KAT
  - HMAC SHA-256 KAT
  - PRNG KAT (FIPS 186-2)

For Libeay.dll:

- Software integrity check (using DSA)

- Know Answer Tests (KATs)
  - AES KAT (ECB, CBC, modes 128/256 bit)
  - HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, and HMAC SHA-512 KATs
  - PRNG KAT – Appendix A.2.4 of ANSI X9.31
  - RSA KAT (encrypt/decrypt and sign/verify)
  - SHA-1 KAT
  - TDES KAT (ECB mode)

The ACE Encryption Engine performs the following conditional self-tests:

For VMCryptolib.dll:

- CRNGT for FIPS 186-2 Appendix 3.1 PRNG
- CRNGT for entropy gathering for FIPS 186-2 Appendix 3.1 PRNG

For Libeay.dll:

- CRNGT for ANSI X9.31 Appedix A.2.4 PRNG
- CRNGT for entropy gathering for ANSI X9.31 Appendix A.2.4 PRNG
- RSA pairwise Consistency Test

The Status Output from the self-tests is stored in RAM[1] (unless the Crypto Officer configures the module to record the Status Output into a local file), and it can be accessed whenever the CO needs to troubleshoot a startup failure.

2.8 Design Assurance

Microsoft Visual SourceSafe (VSS) version 6.0 is used to provide configuration management for the ACE Encryption Engine's FIPS documentation. This software provides access control, versioning, and logging. VSS also maintains an internal revision history of each of the module's files and uniquely labels these revisions.

The VMware source code is maintained in Perforce, which maintains an internal revision history of the module's source files. The Perforce server manages the master file repository that contains every revision of every file under Perforce control. The server maintains a database to track change logs, user permissions, and which users have which files checked out at any time.

2.9 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 level 1 requirements for this validation.

[1] RAM – random access memory

3 Secure Operation

The ACE Encryption Engine meets Level 1 requirements for FIPS 140-2. The section below describes how to place and keep the module in FIPS-approved mode of operation.

3.1 Crypto Officer Guidance

3.1.1 Initial Setup

The Crypto Officer is responsible for installing/uninstalling, configuring, and managing the module. The ACE Encryption Engine is available on a Compact Disc (CD) to the Crypto Officer using a standard carrier (i.e. FedEx or UPS). The Crypto Officer is responsible for inspecting the CD and its packaging upon receipt for signs of tampering. If evidence of tampering is found, the Crypto Officer should contact the vendor immediately and should not use the module. The software module will be provided to the Crypto Officer by VMware Inc. and its marketing affiliates. The module becomes installed during the process of installing the host application, VMware ACE. With the delivered software, the Crypto Officer also receives detailed documentation on installing, uninstalling and configuring VMware ACE. Before installing the module, the CO must configure the PC for single user mode as instructed below:

To configure the Windows platforms for single user mode, the CO must ensure that all remote guest accounts are disabled in order to ensure that only one operator can log into the OS at a time. Guest accounts can be disabled via the **Users and Passwords** selection on the **Control Panel** window.

Services that should be disabled are:

- Server services
- Terminal services
- Remote registry service
- Remote desktop and remote assistance service

For specific information regarding the Microsoft operating system security and administration procedures, please refer to Microsoft's online technical and product information repository at http://technet.microsoft.com/en-us/default.aspx.

3.1.2 Management

The Crypto Officer should monitor the module's status by regularly checking the Windows Event log. If any irregular activity is noticed or the module is consistently reporting errors, then VMware's customer support should be contacted. The CO is also responsible to monitor that FIPS mode of operation is maintained by using only FIPS approved functions. The module should not be used for any FIPS non-approved functions, such as MD5 or DH.

3.1.3 Zeroization

The RSA key pair is erased when the TLS session is terminated. The AES key, TDES key, HMAC keys, and PRNG seed are erased when the session is closed or when the module is reset. The DSA public key is erased when the module is uninstalled.

3.2 User Guidance

The User accesses the module's cryptographic functionalities only. Although the User does not have any ability to modify the configuration of the module, they should check that the host application is enabled and providing cryptographic protection.

The module offers 'Data conversion' service to the Users via the following APIs:

CryptoPass2key_FromString
CryptoPass2key_ToString
CryptoPass2key_Compute
CryptoPass2key_Makekey
Crypto_PasswordWrapData
Crypto_PasswordUnwrapData
Crypto_GetPassword
Crypto_ManglePassphrase
Crypto_EncryptPassword
Crypto_DecryptPassword
Crypto_ClearEncryptedPassword
Crypto_InitializeEncryptedPassword

The keys derived from the APIs mentioned above are not to be used for data encryption/decryption. Otherwise, the resultant values will be considered as plaintext in FIPS mode of operation.

4 Acronyms

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CD	Compact Disc
CD-ROM	Compact Disc Read-Only Memory
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSP	Critical Security Parameter
DLL	Dynamic-Link Library
DSA	Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
OS	Operating System
PC	Personal Computer
PRNG	Pseudorandom Number Generator
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest Shamir and Adleman
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
USB	Universal Serial Bus
VSS	Visual SourceSafe