VMware, Inc.

PCoIP® Cryptographic Module for VMware View™

Software Version: 3.5.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS SECURITY LEVEL: 2
DOCUMENT VERSION: 5

vmware®

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the PCoIP Cryptographic Module for VMware View from VMware, Inc. This Security Policy describes how the PCoIP Cryptographic Module for VMware View (software version: 3.5.0) meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module.

Federal Information Processing Standards (FIPS) Publication 140-2 – *Security Requirements for Cryptographic Modules* details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the Cryptographic Module Validation Program (CMVP) website (http://csrc.nist.gov/groups/STM/cmvp), which is maintained by National Institute of Standards and Technology (NIST) and Communication Security Establishment of Canada (CSEC).

The PCoIP Cryptographic Module for VMware View 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 corporate website (http://www.vmware.com) contains information on the full line of products from VMware.
- The CMVP Validation List (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) 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:

- Submission Summary
- Vendor Evidence document
- Finite State Model
- Other supporting documentation and 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 PCoIP Cryptographic Module for VMware View

2.1 Overview

VMware, Inc. is a global leader in virtualization and cloud infrastructure, delivering customer-proven solutions that reduce IT[1] complexity and accelerate the transition to a cloud computing approach, all while preserving existing investments and improving security and control.

VMware View is a desktop virtualization platform, and the only purpose-built solution for delivering desktops as a secure managed service. VMware View virtualizes desktop operating systems, applications, and user data, decoupling these components from their underlying hardware. IT administrators can then centrally manage and provision these desktop components from the datacenter, creating a rapidly-deployable private "desktop cloud" infrastructure.

VMware View employs the PC[2]-over-IP[3®] (PCoIP) protocol, designed by Teradici Corporation specifically to deliver a user's desktop from a centralized host PC over standard IP networks. The PCoIP protocol compresses, encrypts, and encodes the entire computing experience at the datacenter and transmits all required information across a standard IP network to PCoIP client devices. The PCoIP protocol facilitates the central provisioning, management, maintenance, and securing of enterprise desktops.

VMware's software implementation of the PCoIP protocol consists of a PCoIP Client and PCoIP Server. PCoIP Client installs on existing PCs or thin clients across the network, while PCoIP Server installs on the server or VMware virtual machine with no special hardware required. The communication between the PCoIP Client and PCoIP Server is secured via the PCoIP Cryptographic Module. Using the PCoIP Crypto Module, the session management traffic is protected via the TLS[4] protocol with FIPS-Approved algorithms, while the bulk data traffic is protected via the PCoIP protocol which encrypts and authenticates all the data sent over the network using AES-GCM[5] authenticated encryption algorithm.

In a typical deployment, software-based clients and servers are used to remotely access PC desktops. Figure 1 shows a typical deployment scenario in which software-based PCoIP clients are used to access virtual machine desktops.

[1] IT – Information Technology
[2] PC – Personal Computer
[3] IP – Internet Protocol
[4] TLS – Transport Layer Security
[5] GCM – Galois Counter Mode

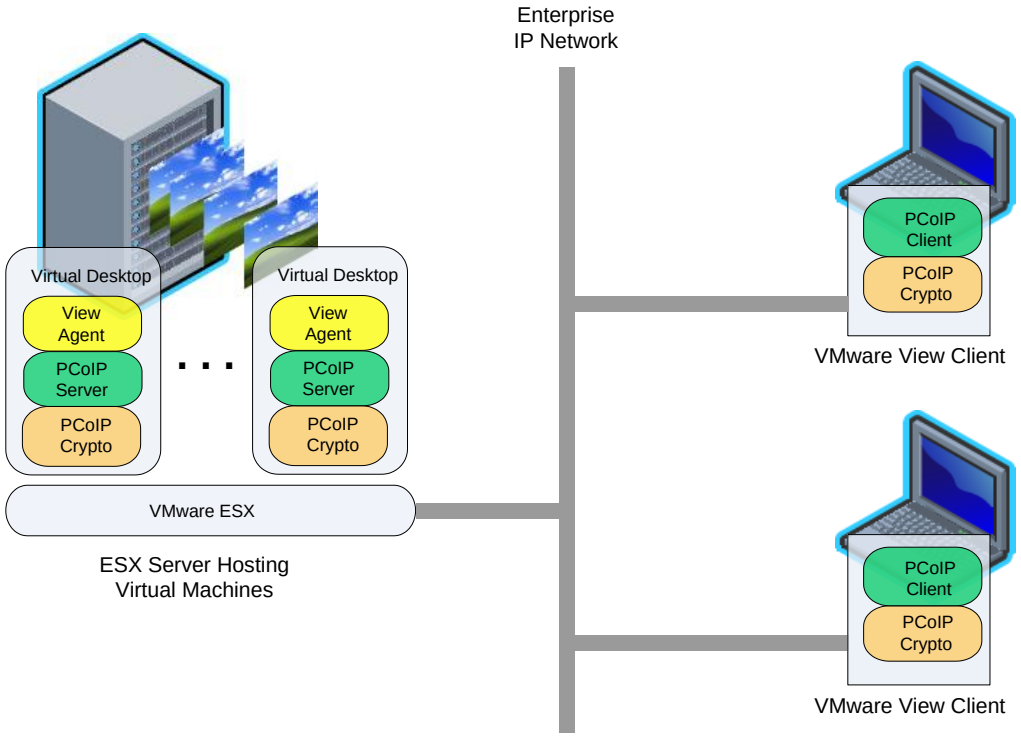

Figure 1 – Typical Deployment of PCoIP System

All of the cryptographic functionality performed by the VMware View software within a PCoIP session is provided by the PCoIP Cryptographic Module for VMware View. The PCoIP Cryptographic module is a single shared library which is used by both the PCoIP server and the PCoIP client applications. The module is a multi-chip standalone cryptographic module evaluated for use on a standard General Purpose Computer (GPC) platform running one of the following operating systems:

- Windows XP
- Red Hat Enterprise Linux (RHEL) 5.1

The PCoIP® Cryptographic Module for VMware View is validated as a multi-chip standalone module at the following FIPS 140-2 Section levels:

Table 1 – FIPS 140-2 Security Levels

Section	Section Title	Level
1	Cryptographic Module Specification	2
2	Cryptographic Module Ports and Interfaces	2
3	Roles, Services, and Authentication	3
4	Finite State Model	2
5	Physical Security	N/A
6	Operational Environment	2

Section	Section Title	Level
7	Cryptographic Key Management	2
8	EMI/EMC[6]	2
9	Self-tests	2
10	Design Assurance	2
11	Mitigation of Other Attacks	N/A

2.2 Cryptographic boundary

The following sections will define the physical and logical boundary of the PCoIP Cryptographic Module.

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module there are no physical security mechanisms implemented per se, the module must rely on the physical characteristics of the GPC. Thus, the physical cryptographic boundary of the PCoIP Cryptographic Module for VMware View is defined by the hard metal enclosure around the computer on which it runs. Figure 2 below depicts the standard hardware platform with accompanying physical ports, the dotted blue line surrounding the module components represents the module's physical cryptographic boundary, while the ports and interfaces exist at the boundary and interface with the various controllers.

[6] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

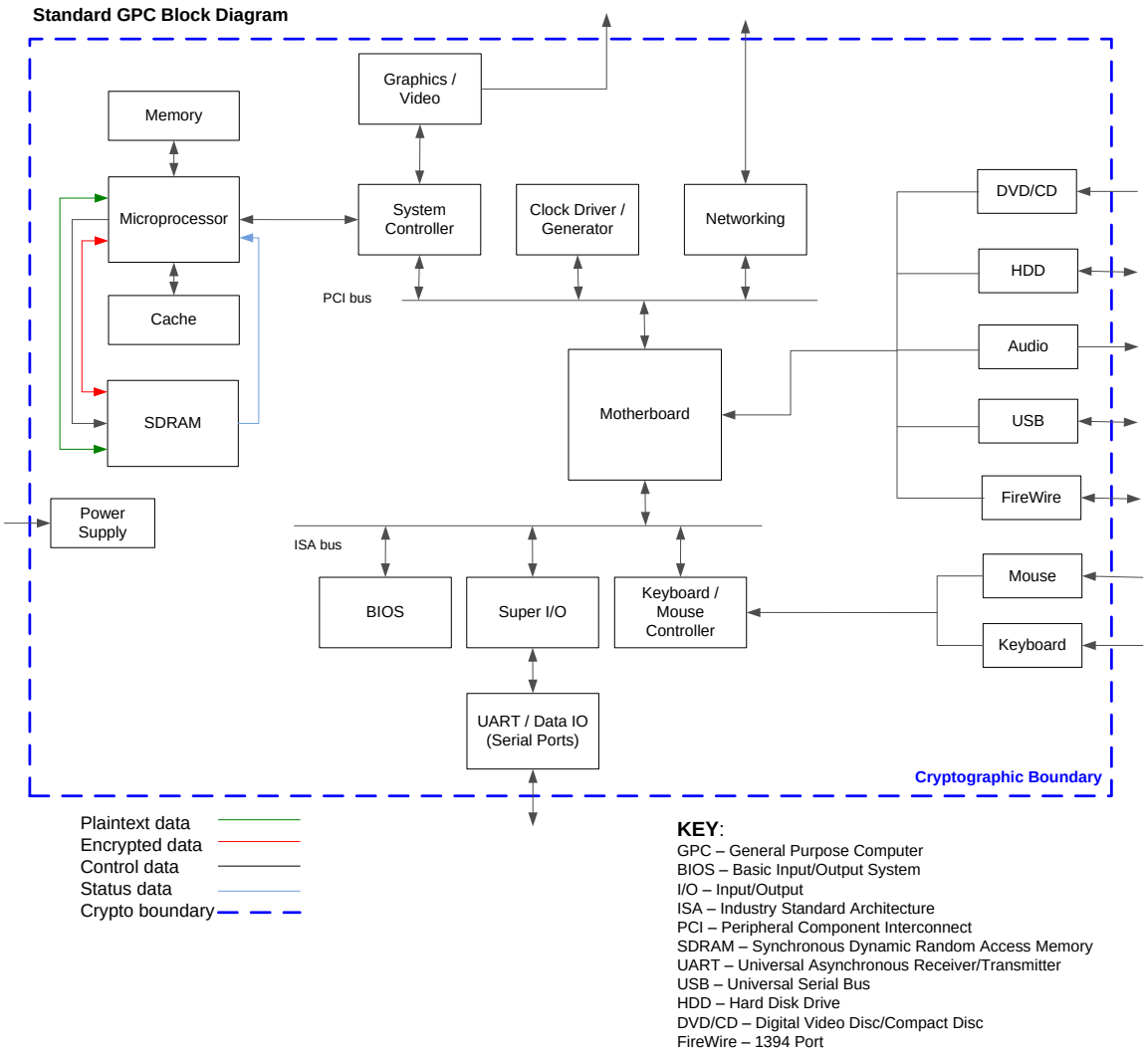

Figure 2 – Standard GPC Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 3 below shows a logical block diagram of the module, where "Client Application" can represent either the PCoIP Client or PCoIP Server application described in Section 2.1 above. The module runs on Windows XP or RHEL 5.1 Operating System. The module's logical cryptographic boundary encompasses all functionality contained within a single shared library named pcoip_crypto.dll (on Windows XP) or libpcoip_crypto.so (on RHEL 5.1).

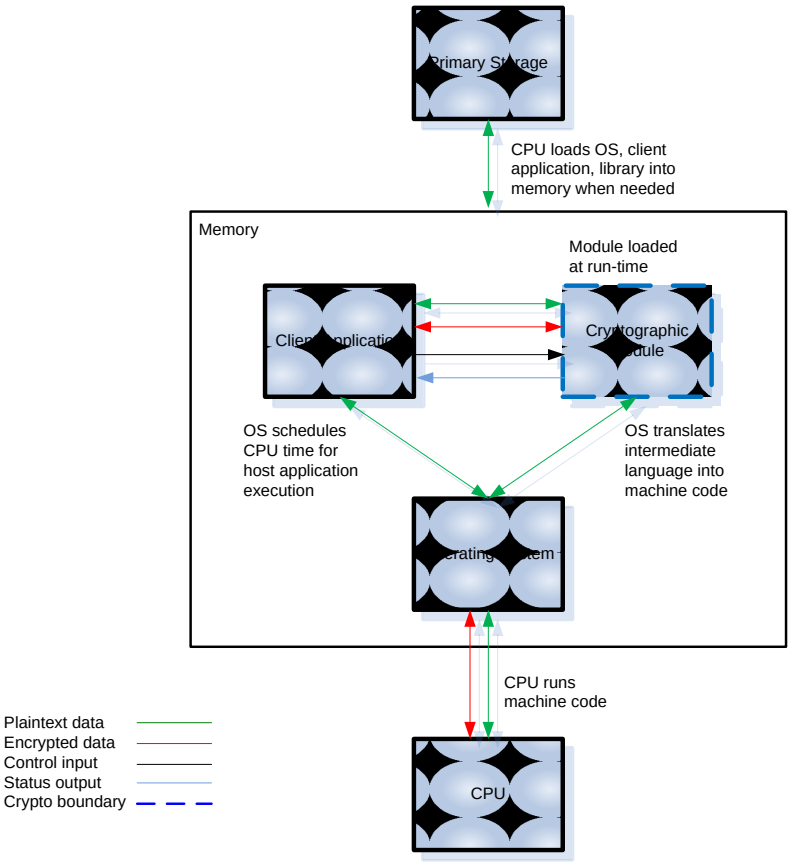

Figure 3 – Logical Block Diagram and Cryptographic Boundary

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 five logical interfaces:

- Data input
- Data output
- Control input
- Status output
- Power

The module features the physical ports of the GPC, as depicted in Figure 2. The following is a list of physical interfaces implemented on a GPC:

- Keyboard port
- Network port
- Mouse port
- Audio port
- Graphics/Video port
- DVD/CD[7] drive
- LED[8] indicators

[7] DVD/CD – Digital Video Disc/Compact Disc

- Power plug/adapter
- Serial port
- Power switch
- USB ports
- FireWire Port

As a software module, the module has no physical characteristics. Thus, the module's manual controls, physical indicators, and physical and electrical characteristics are those of the GPC. The FIPS-defined interfaces map to their physical and logical counterparts as described in Table 2 below.

Table 2 – FIPS Interface Mappings

FIPS-Defined Logical Interface	Physical Interface Mapping (GPC)	Logical Interface Mapping (Module)
Data Input Interface	Keyboard, mouse, DVD/CD, and serial/network/USB/FireWire port	The API calls that accept input data for processing through their arguments.
Data Output Interface	Graphics/Video, audio and serial/ /network/USB/FireWire port	The API calls that return by means of their return codes or arguments generated or processed data back to the caller.
Control Input Interface	Keyboard, mouse, DVD/CD, and serial/network/USB/FireWire port, power switch	The API calls that are used to initialize and control the operation of the module.
Status Output Interface	Graphics/Video, audio, LED indicators and serial/network/USB/ FireWire port	Return values for API calls.
Power Interface	Power input	Not Applicable

2.4 Roles and Services

The module supports the following authorized roles: the Crypto-Officer (CO) role and the User role. Once authenticated, the operator is authorized to assume both the Crypto-Officer and User role.

Descriptions of the services available to the Crypto-Officer and User role are listed in Table 3 below. The following definitions are used in the "CSP[9] and Type of Access" column in Table 3:

- **R** – The CSP is **read** or **referenced** by the service.
- **W** – The CSP is **written** or **updated** by the service.
- **X** – The CSP is used in the **execution** of a cryptographic function.

Table 3 – Mapping of Crypto-Officer and User Role's Services to Type of Access

Service	Role		Description	CSP and Type of Access
	CO	User		
tera_crypto_aes_256_encrypt	✓	✓	Performs AES-256 ECB encryption	AES ECB Key – R, W, X
tera_crypto_aes_256_decrypt	✓	✓	Performs AES-256 ECB decryption	AES EBC Key – R, W, X

[8] LED – Light-Emitting Diode
[9] CSP – Critical Security Parameter

Service	Role		Description	CSP and Type of Access
	CO	User		
tera_crypto_cipher_create()	✓	✓	Creates a cipher object	AES Key – R, W, X IV – R, W, X Legacy[10] AES Key – R, W, X
tera_crypto_cipher_csps_get()	✓	✓	Gets the key, salt, and SPI for the specified cipher object	AES Key – R IV – R
tera_crypto_cipher_csps_set()	✓	✓	Sets the key, salt, and SPI for the specified cipher object	AES Key – R, W IV – R, W
tera_crypto_cipher_csps_use_legacy()	✓	✓	Specifies whether to use the legacy salt, key, and SPI or encrypted key, salt, and SPI when encrypting or decrypting with the cipher object	AES Key – R, W IV – R, W Legacy AES Key – R, W
tera_crypto_cipher_delete()	✓	✓	Zeroizes all key, salt, and SPI values and returns cipher objects to the memory pool	AES Key – R, W Legacy AES Key – R, W
tera_crypto_cipher_legacy_key_get()	✓	✓	Gets the key for the specified cipher object	Legacy AES Key – R
tera_crypto_cipher_legacy_key_set()	✓	✓	Sets the key for the specified cipher object	Legacy AES Key – R, W
tera_crypto_cipher_legacy_salt_get()	✓	✓	Gets the salt for the specified cipher object	None
tera_crypto_cipher_legacy_salt_set()	✓	✓	Sets the salt for the specified cipher object	None
tera_crypto_cipher_legacy_spi_get()	✓	✓	Gets the SPI for the specified cipher object	None
tera_crypto_cipher_legacy_spi_set()	✓	✓	Sets the SPI for the specified cipher object	None
tera_crypto_cipher_spi_get()	✓	✓	Gets the SPI for the specified cipher object	None
tera_crypto_cipher_spi_set()	✓	✓	Sets the SPI for the specified cipher object	None
tera_crypto_esp_packet_authenticate()	✓	✓	Authenticates a received IPsec ESP packet	None
tera_crypto_esp_packet_decrypt()	✓	✓	Decrypts the ESP[11] packet using the specified cipher object	AES Key – R, W, X, IV – R, W, X Legacy AES Key – R, W, X AES ECB[12] Key – R, W, X

[10] Legacy – Unencrypted AES GCM Parameters
[11] ESP – Encapsulating Security Payload
[12] ECB – Electronic Codebook

Service	Role		Description	CSP and Type of Access
	CO	User		
tera_crypto_esp_packet_encrypt()	✓	✓	Encrypts the ESP packet using the specified cipher object	AES Key – R, W, X, IV – R, W, X Legacy AES Key – R, W, X AES ECB Key – R, W, X
tera_crypto_esp_packet_handle_get()	✓	✓	Retrieves the cipher object handle	None
tera_crypto_exit()	✓	✓	Shuts down the cryptographic module (zeroizing keys) before program exit	AES Key – R, W IV – R, W, X Legacy AES Key – R, W
tera_crypto_fips_mode_get()	✓	✓	Returns the current FIPS mode of operation	None
tera_crypto_fips_mode_set()	✓	✓	Enables or disables the FIPS mode of operation	None
tera_crypto_get_build_id()	✓	✓	Retrieves the build ID (version) of the crypto module	None
tera_crypto_get_build_date()	✓	✓	Retrieves the date that the crypto module was build	None
tera_crypto_init()	✓	✓	Initializes the cryptographic module at the start of the application	None
tera_crypto_rand_bytes()	✓	✓	Calls ANSI X9.31 RNG to create a cryptographically strong pseudo-random number	Seed key – R, W, X Seed – R, W, X IV – R, W, X
tera_crypto_self_test()	✓	✓	Performs the FIPS power-up self test	HMAC SHA-256 Key – R, W, X
Send and receive session initiation data	✓	✓	OpenSSL used for TLS connection for management traffic	AES CBC Key – R, W, X Seed – R, W, X Seed key – R, W, X

2.4.1 Authentication Mechanism

The module employs the following authentication method to authenticate the Crypto-Officer and the User.

Table 4 – 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 2×10^{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 2×10^{12} (100 in 2×10^{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 PCoIP® Cryptographic Module for VMware View is purely a software module. As such, it depends on the physical characteristics of the GPC and its protection mechanisms. Thus, physical security requirements do not apply.

2.6 Operational Environment

The module was tested for FIPS 140-2 validation running on Windows XP and Red Hat Enterprise Linux (RHEL) 5.1 operating systems. All cryptographic keys and CSPs are either hardcoded or under the control of the OS. The OS protects the CSPs against unauthorized disclosure, modification, and substitution. The OS uses its native memory management 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.

The required operating systems on which the module is supported are Windows XP and RHEL 5.1, which were evaluated to the CC[13] requirements at evaluation assurance level 4+. Windows XP was validated on 01 April 2007 (Validation Report Number: CCEVS-07-0023, http://www.niap-ccevs.org/st/st_vid9506-vr.pdf) and RHEL 5.1 was validated on 21 April 2008 (Validation Report Number: CCEVS-VR-VID10286-2008, http://www.niap-ccevs.org/st/st_vid10286-vr.pdf).

2.7 Cryptographic Key Management

The module employs the FIPS-Approved algorithms listed in Table 5 below.

[13] CC – Common Criteria

Table 5 – FIPS-Approved Algorithm Implementations

Approved Security Function	Certificate Number
Symmetric Key Algorithm	
AES GCM – 128-bit	1640
AES CBC[14] – 128- and 256-bit	1642
AES ECB – 256-bit	1639
Secure Hash Standard (SHS)	
SHA-256*	1443
Message Authentication Code (MAC) Function	
HMAC* using SHA-256	964
Pseudo Random Number Generator (PRNG)	
ANSI[15] X9.31 Appendix A.2.4 PRNG*	879

Additionally, the module utilizes the following non-FIPS-Approved algorithm implementations:

- Salsa12: 256-bit
- RSA encrypt/decrypt (key transport): 1024-, 2048-, 3072-, 7680-, 15360-bits (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength).*

[14] CBC – Cipher Block Chaining
[15] ANSI – American National Standards Institute
*see SP800-131A for deprecation statements

The module supports the following critical security parameters:

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	CSP Strength	Generation / Input	Output	Storage	Zeroization	Use
AES Key	128-bit AES GCM key	128-bit	Generated internally using the ANSI X9.31 PRNG or entered electronically in encrypted form	Exits the module in encrypted form	Stored in volatile memory	By power cycle or Zeroization Service	Symmetric encryption/decryption (for data plane) using 128-bit AES GCM
Initialization Vector (IV)	64-bit CSP	64-bit	Generated internally using the ANSI X9.31 PRNG	Never exits the module	Stored in volatile memory	By uninstalling the module	Symmetric encryption/decryption (for data plane) using 128-bit AES GCM
RSA Public Key	RSA 1024- to 15360-bit key	Between 80- to 256-bits	Hard coded in module binary	Exits the module in plaintext form	Stored in module binary	By power cycle or Zeroization Service	Key Transport
RSA Private Key	RSA 1024- to 15360-bit key	Between 80- to 256-bits	Hard coded in module binary	None	Stored in module binary	By uninstalling module and reformatting the GPC hard drive	Key Transport
Legacy AES Key	128-bit AES GCM key	128-bit	Generated internally using the ANSI X9.31 PRNG or entered electronically in plaintext form	Exits the module in plaintext form	Stored in volatile memory	By power cycle or Zeroization Service	Symmetric encryption/decryption (for data plane) using 128-bit AES GCM

CSP	CSP Type	CSP Strength	Generation / Input	Output	Storage	Zeroization	Use
AES ECB Key	256-bit AES ECB key	256-bit	Generated during development and hard-coded in the module	Never exits the module	Hard-coded	By uninstalling the module and reformatting the GPC hard drive	Encryption of AES key, salt, and SPI using 256-bit AES ECB for key wrapping
AES CBC Key	128-, 256-bit AES CBC key	128-, 256-bit	Generated internally using the ANSI X9.31 PRNG or entered electronically in encrypted form	None	Stored in volatile memory	By power cycle or TLS session termination	Symmetric encryption/decryption
HMAC SHA-256 Key	HMAC SHA-256	112-bit	Generated during development and hard-coded in the module	None	Hard-coded	By uninstalling the module and reformatting the GPC hard drive	Perform software integrity test
X9.31 PRNG Seed Key	128, 256-bit AES key	128, 256-bit	Generated internally	None	Stored in volatile memory	By power cycle or TLS session termination	Generate FIPS Approved random number
X9.31 PRNG Seed	128-bit of Seed value	128-bit	Generated internally	None	Stored in volatile memory	By power cycle or TLS session termination	Generate 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.

2.7.2 Key Entry and Output

Keys can be passed to the cryptographic module as parameters from the client application via the APIs. The client application using the module is responsible for ensuring that the input of secret and private keys is accomplished in encrypted form. The legacy AES key and the RSA public key are the only keys that can be output from the module in plaintext. The legacy key never leaves the physical boundary. It is also possible to input the legacy AES key in plaintext.

2.7.3 CSP Storage and Zeroization

The module persistently stores the following keys and CSPs:

- AES ECB Key
- HMAC SHA-256 Key
- RSA Private Key

All persistently-stored keys can be zeroized by uninstalling the cryptographic module and reformatting the hard drive. All ephemeral keys and CSPs can be zeroized by calling the tera_crypto_cipher_delete() or tera_crypto_exit() API described in Table 6, or by power-cycling the host GPC. The AES CBC key, X9.31 seed key, and X9.31 seed are OpenSSL keys and are zeroized by the termination of the TLS connection.

2.8 EMI/EMC

The module is a software module and depends on the GPC for its physical characteristics. However, the GPC must 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 systems sold in the United States must meet the applicable FCC requirements.

2.9 Self-Tests

The module performs the following self-tests at power-up:

- Software integrity test using HMAC SHA-256
- Cryptographic Algorithm tests
  - AES GCM KAT[16]
  - AES ECB encrypt/decrypt KAT
  - AES CBC encrypt/decrypt KAT
  - HMAC SHA-256 KAT
  - RSA encrypt/decrypt KAT
  - ANSI X9.31 PRNG Appendix A.2.4 KAT

The module performs the following conditional self-tests:

- ANSI X9.31 Continuous Random Number Generator test (CRNGT)

The module enters an error state when a power-up self-test fails. The power-up self-tests run through to completion without interruption, and provide no mechanism for data output. Upon any power-up self-test failure, the module enters a critical error state where it sets an internal error flag and returns the control back to the client application. A power-up self-test error can only be cleared by restarting the module.

[16] KAT – Known Answer Test

Failure of a conditional self-test transitions the module to a critical error state. No data output or cryptographic operations are possible when the module enters this state. After the module provides error status, the critical error can only be cleared by restarting the module.

2.10 Design Assurance

VMware uses Subversion (SVN) version control system as the configuration management system. The control system provides code version control, code sharing and build management. SVN supports source code check-in, check-out, and merging. Branches in subversion are used to isolate projects and release developments. It can also be locked during different phases of control changes.

Additionally, Microsoft Visual SourceSafe (VSS) version 6.0 is used to provide configuration management for the module's FIPS documentation. This software provides access control, versioning, and logging.

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 PCoIP® Cryptographic Module for VMware View meets Level 2 requirements for FIPS 140-2. The sections below describe how to place and keep the module in its FIPS-Approved mode of operation.

3.1 Modes of Operation

The cryptographic module will be provided as a part of the client application. The module is a shared library that is accessed by a client application to provide cryptographic services. The module has two modes of operation: FIPS-Approved mode and non-FIPS-Approved mode.

Non-FIPS-Approved services are disabled in FIPS-Approved mode of operation. In its non-FIPS-Approved mode of operation, the module supports Salsa12-256 algorithm. The non-FIPS-Approved mode of operation is the default configuration.

3.2 Status Monitoring

Operators are able to monitor the status of the module by using the tera_crypto_fips_mode_get() service. The Crypto-Officer should monitor the module's status regularly for FIPS mode of operation. The CO can also check the status of the module by checking the log file of the host operating system. Statuses are represented as follows:

- "Running in the FIPS approved mode"
- "Running in the non-FIPS mode"
- "FIPS power-up/self-test passed"
- "FIPS power-up/self-test failed"

3.3 Crypto-Officer Guidance

With the delivered client application software, the CO receives detailed documentation on installing, uninstalling, configuring, managing, and upgrading the client application.

3.3.1 Installation

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 client application. The CO must ensure that the client application is installed on one of the CC-evaluated operating systems listed in Section 2.6. The CO must also ensure that the client application is installed on a machine that meets the minimum hardware and software requirements.

3.3.2 Configuration

The module's mode of operation is determined by a configuration setting. Full instructions for the CO to place the module in FIPS-Approved mode are found in the document *Enabling the FIPS-Approved Mode of Operation in the VMware View Rev 1.0*. Once the VMware View is configured to operate in the FIPS-Approved mode, the PCoIP Cryptographic module starts in the FIPS-Approved mode of operation and only uses FIPS-Approved ciphers for all cryptographic functionality. If the VMware View is not configured for the FIPS-Approved mode, then the module starts in non-FIPS-Approved mode and it may use non-FIPS-Approved ciphers as described in section 3.1.

Once the mode of operation is set, it can only be changed by calling tera_crypto_exit(), followed by a call to tera_crypto_init().

3.3.3 Management

No additional management activities are required for the secure operation of the module.

3.4 User Guidance

The User does not have any ability to install the module. Operators in the User role are able to use the services available to the User role listed in Table 3. However, they should report to the Crypto-Officer if any irregular activity is noticed.

4 Acronyms and Terms

This section defines the acronyms and terms used throughout the Security Policy.

Table 7 – Acronyms

Acronym/Term	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
BIOS	Basic Input/output System
CBC	Cipher-Block Chaining
CC	Common Criteria
CD	Compact Disc
CMVP	Cryptographic Module Validation Program
CO	Crypto-Officer
CPU	Central Processing Unit
CRNGT	Continuous Random Number Generator Test
CSEC	Communications Security Establishment of Canada
CSP	Critical Security Parameter
DVD	Digital Video Disc
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESP	Encapsulating Security Payload
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
GPC	General Purpose Computer
GPO	Group Policy Object
HD	High Definition
HDD	Hard Disk Drive
HMAC	(Keyed-) Hash Message Authentication Code
I/O	Input/Output
IP	Internet Protocol
ISA	Industry Standard Architecture
IT	Information Technology

Acronym/Term	Definition
KAT	Known Answer Test
LED	Light Emitting Diode
Legacy	Unencrypted AES GCM Parameters
NIST	National Institute of Standards and Technology
OS	Operating System
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCoIP	Personal Computer Over Internet Protocol
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
RSA	Rivest Shamir and Adleman
Salsa	Stream Cipher
SDRAM	Synchronous Dynamic Random Access Memory
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SPI	Security Parameter Index
SVN	Subversion
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VSS	Visual SourceSafe

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com