VMware, Inc.

3401 Hillview Ave
Palo Alto, CA 94304, USA
Tel: 877-486-9273
Email: info@vmware.com
http://www.vmware.com

VMware's IKE Crypto Module

Software Version: 1.1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.8

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

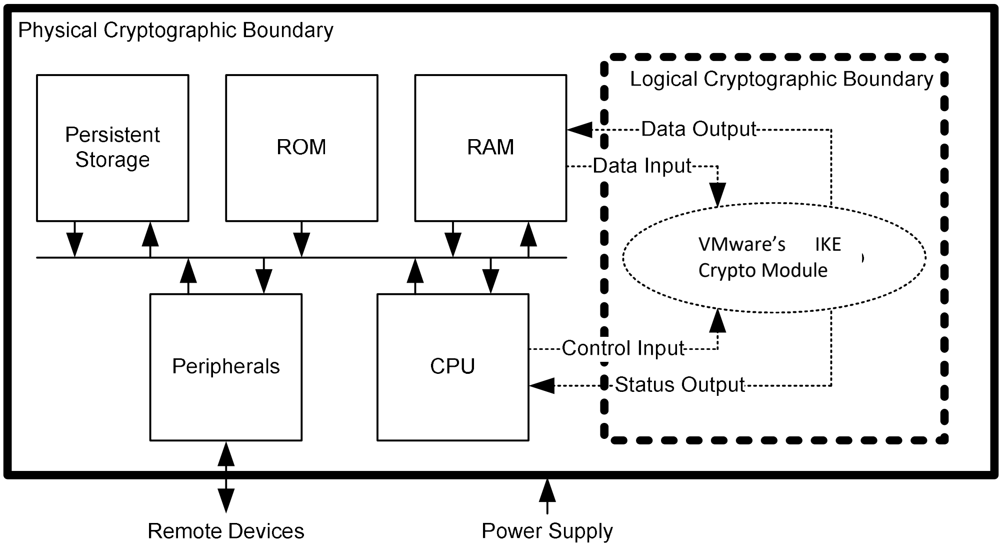
1 Introduction

VMware's IKE Crypto Module is a FIPS 140-2 Security Level 1 validated software cryptographic module. This module is a toolkit that provides the most commonly used cryptographic primitives for a wide range of applications, including primitives needed for VPN (Virtual Private Network), TLS (Transport Layer Security), DAR (Data-At-Rest), and DRM (Digital Rights Management) clients.

VMware's IKE Crypto Module is a software-based product with a custom, small-footprint API (Application Programming Interface). The cryptographic module has been designed to provide the necessary cryptographic capabilities for other VMware products. However, it can also be used stand-alone in custom-developed products to provide the required cryptographic functionality.

The module is primarily intended for embedded products with a general-purpose operating system.

Figure 1 – VMware's IKE Crypto Module Cryptographic Boundary



For FIPS 140-2 purposes, VMware's IKE Crypto Module is classified as a multi-chip standalone cryptographic module. Within the *logical* boundary of VMware's IKE Crypto Module is the `libsafezone-sw-fips.a/so` object code library. The *physical* cryptographic boundary of the module is the enclosure of a general-purpose computing device executing the application that embeds the VMware's IKE Crypto Module.

The VMware's IKE Crypto Module has been tested for validation on the following platforms:

Table 1 – Tested Configuration

Operating System	Processor	Optimization	Hypervisor	Hardware
PhotonOS 2.0	Intel Xeon 6126	AES-NI	ESXi 6.7	Dell PowerEdge R740
PhotonOS 2.0	Intel Xeon 6126	None	ESXi 6.7	Dell PowerEdge R740
Ubuntu 16.04	Intel Xeon 6126	AES-NI	ESXi 6.7	Dell PowerEdge R740
Ubuntu 16.04	Intel Xeon 6126	None	ESXi 6.7	Dell PowerEdge R740
VMware SD-WAN OS 3.3	Intel Xeon 6126	AES-NI	ESXi 6.7	Dell PowerEdge R740

VMware SD-WAN OS 3.3	Intel Xeon 6126	None	ESXi 6.7	Dell PowerEdge R740
VMware SD-WAN OS 3.3	Intel Atom C3308	AES-NI	-	VMware SD-WAN Edge 610
VMware SD-WAN OS 3.3	Intel Atom C3308	None	-	VMware SD-WAN Edge 610
VMware SD-WAN OS 3.3	Intel Xeon D-2187NT	AES-NI	-	VMware SD-WAN Edge 3800
VMware SD-WAN OS 3.3	Intel Xeon D-2187NT	None	-	VMware SD-WAN Edge 3800
Ubuntu 16.04	Intel Xeon Gold 6126	AES-NI	ESXi 7.0	Dell PowerEdge R740
Ubuntu 16.04	Intel Xeon Gold 6126	None	ESXi 7.0	Dell PowerEdge R740
Ubuntu 18.04	Intel Xeon Gold 6126	AES-NI	ESXi 7.0	Dell PowerEdge R740
Ubuntu 18.04	Intel Xeon Gold 6126	None	ESXi 7.0	Dell PowerEdge R740
VMware SD-WAN OS 4.0	Intel Xeon Gold 5218	AES-NI	ESXi 7.0	Dell PowerEdge R640
VMware SD-WAN OS 4.0	Intel Xeon Gold 5218	None	ESXi 7.0	Dell PowerEdge R640

Compliance is maintained on platforms for which the binary executable remains unchanged. The module has been confirmed by the vendor to be operational on the following platforms. As allowed by the FIPS 140-2 Implementation Guidance G.5, the validation status of the Cryptographic Module is maintained when operated in the following additional operating environments:

- VMware SD-WAN Edge 510
- VMware SD-WAN Edge 510-LTE-NAM-EMEA
- VMware SD-WAN Edge 510-LTE-APAC
- VMware SD-WAN Edge 520
- VMware SD-WAN Edge 520v
- VMware SD-WAN Edge 540
- VMware SD-WAN Edge 840
- VMware SD-WAN Edge 2000
- VMware SD-WAN Edge 610
- VMware SD-WAN Edge 620
- VMware SD-WAN Edge 640
- VMware SD-WAN Edge 680
- VMware SD-WAN Edge 3400
- VMware SD-WAN Edge 3800
- VMware SD-WAN Edge 3810
- VMware SD-WAN Virtual Edge
- VMware SD-WAN OS

Further, VMware, Inc. affirms that the VMware's IKE Crypto Module runs in its configured, Approved mode of operation on the following binary compatible platforms executing VMware ESXi 6.5, ESXi 6.7, ESXi 7.0, or ESXi 8.0 with any of the above listed OS along with Ubuntu 18.04, Ubuntu 18.10, Ubuntu 20.04, SD-WAN OS 4.x, Photon OS 2, Photon OS 3, or Photon OS 4:

- Dell PowerEdge T320, R530, R640, R650, R730, and R830 with Intel Xeon Processor
- Dell PowerEdge R740 Gen14 with Intel Xeon Processor
- HPE ProLiant DL380 Gen9 with Intel Xeon Processor
- HPE ProLiant DL38P Gen8 with AMD Opteron Processor
- Cisco UCS – B22 M Series Blade Servers with Intel Processor
- Cisco UCS – C24 M3 Series Rackmount with Intel Xeon Processor
- A general-purpose computer platform with an Intel Atom, Intel Core I, Intel Xeon, or AMD Opteron Processor executing VMware ESXi and any OS (including Apple OS (OS X, macOS, iOS), Android OS, OpenWrt, VMware SD-WAN OS, and others) with single user mode.
- A cloud computing environment composed of a general-purpose computing platform executing VMware ESXi or a VMware cloud solution that is executing VMware ESXi.

The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when the specific operational environment is not listed on the validation certificate.

1.1 Purpose

The purpose of this document is to describe the secure operation of the VMware's IKE Crypto Module including the initialization, roles, and responsibilities of operating the product in a secure, in FIPS 140 mode of operation.

1.2 Security level

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2.

Table 2 – Security Level Per FIPS 140-2 Section

Security Level	
Security Requirements Specification	**Level**
Cryptographic Module Specification	1
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

1.3 Glossary

Table 3 – Glossary

Term/Acronym	Description
AES	Advanced Encryption Standard
API	Application Programming Interface
CMVP	Cryptographic Module Validation Program (FIPS 140)
CSP	Critical Security Parameter
DEP	Default Entry Point
DRM	Digital Rights Management
DSS	Digital Signature Standard
EC	Elliptic Curve
FIPS	Federal Information Processing Standard
IKE	Internet Key Exchange
KEM	Key-Encapsulation Mechanism (See NIST SP 800-56B)
KTS	Key Transport Scheme
OAEP	Optimal Asymmetric Encryption Padding
PRF	Pseudo-Random Function
SHS	Secure Hash Standard
SRDI	Security Relevant Data Item

TLS	Transport Layer Security
Triple-DES	Triple Data Encryption Standard
VPN	Virtual Private Network

2 Ports and Interfaces

As a software-only module, the VMware's IKE Crypto Module provides an API logical interface for invocation of FIPS140-2 approved cryptographic functions. The functions shall be called by the referencing application, which assumes the operator role during application execution. The API, through the use of input parameters, output parameters, and function return values, defines the four FIPS 140-2 logical interfaces: data input, data output, control input and status output.

Table 4 – FIPS 140-2 Logical Interface Mapping

Logical Interfaces	API
Data Input	The data read from memory area(s) provided to the invoked function via parameters that point to the memory area(s).
Control Input	The API function invoked and function parameters designated as control inputs.
Data Output	The data written to memory area(s) provided to the invoked function via parameters that point to the memory area(s).
Status Output	The return value of the invoked API function.
Power Interface	Not accessible via the API. The power interface is used as applicable on the physical device.

3 Roles, Services and Authentication

The VMware's IKE Crypto Module supports the *Crypto Officer* and *User* roles. The operator of the module will assume one of these two roles. Only one role may be active at a time. The Crypto Officer role is assumed implicitly upon module installation, uninstallation, initialization, zeroization, and power-up self-testing. If initialization and self-testing are successful, a transition to the User role is allowed and the User will be able to use all keys and cryptographic operations provided by the module, and to create any CSPs.

The four unique run-time services given only to the Crypto Officer role are the ability to initialize the module, to modify the entropy source, and to switch to the User role to perform any activities allowed for the User role. The VMware's IKE Crypto Module does not support concurrent operators.

3.1 Roles and Services

The module does not authenticate the operator role.

3.1.1 User Role

The User role is assumed once the Crypto Officer role is finished with module initialization and explicitly switches the role using the *FL_LibEnterUserRole* API function. The User role is intended for common cryptographic use. The full list of cryptographic services available to the User role is supplied in chapter 5 of this document.

Table 5 – Services for User Role

Service	Description
All services except installation, initialization, entropy source nomination.	All standard cryptographic operations of the module, such as symmetric encryption, message authentication codes, and digital signatures. The User role may also allocate the key assets and load values for any of these cryptographic purposes. The VMware's IKE Crypto Module also provides a 'Show Status' service (API function `FL_LibStatus`) that can be used to query the current status of the cryptographic module. A macro based on `FL_LibStatus` is provided (`FL_IS_IN_APPROVED_MODE`), which returns true if the module is currently in an approved mode of operation.

3.1.2 Crypto-officer Role

The Crypto Officer role can perform all the services allowed for the User role plus a handful of additional ones. Separate from the run-time services of the module, the tasks of installing and uninstalling the module to and from the host system imply the role of a Crypto Officer. The four run-time services available only to the Crypto Officer are initializing the module for use, modifying the entropy source, and switching to the User role.

Table 6 – Services for Crypto-officer Role

Service	Description
All services allowed for User role	See above.
Initialization	Loading and preparing the module for use.
Entropy Source	Select the provider of the external entropy source. (`FL_RbgInstallEntropySource`, `FL_RbgRequestSecurityStrength`), (`Non-Approved & Disallowed: FL_RbgUseNonblockingEntropySource`).
Switch to the User Role	Uses the `FL_LibEnterUserRole` API function to switch to User role.
Installation	When the module is installed to a host system.
Uninstallation	When the module is removed from a host system.

3.2 Authentication Mechanisms and Strength

FIPS 140-2 Security Level 1 does not require *role-based* or *identity-based* operator authentication. The VMware's IKE Crypto Module will not authenticate the operator.

4 Security Operation and Security Rules

In order to operate the VMware's IKE Crypto Module securely, the operator should be aware of the security rules enforced by the module and should adhere to the rules for physical security and secure operation.

4.1 Security Rules

To operate the VMware's IKE Crypto Module securely, the operator of the module must follow these instructions:

1. The operating environment that executes the VMware's IKE Crypto Module must ensure single operator mode of operation to be compliant with requirements for the FIPS 140-2 Level 1.
2. The correct operation of the module depends on the Default Entry Point. It is not allowed to prevent execution of the Default Entry Point (the function `FL_LibInit`).
3. The operator must not call `ptrace` or `strace` functions or run `gdb` or other debugger when the module is in the FIPS mode.
4. If the hardware platform has a connector for an external debugger (for example JTAG), that connector must not be used while the module is in FIPS mode.
5. The VMware's IKE Crypto Module keeps all CSPs and other protected objects in Random Access Memory (RAM). The operator(s) must only use these objects via the handles provided by the VMware's IKE Crypto Module. It is not permissible to directly access these objects in the memory.
6. The operator must not call functions provided by the VMware's IKE Crypto Module that are not explicitly specified in the appropriate guidance document for User or Crypto Officer.
7. When using cryptographic services provided by the VMware's IKE Crypto Module, the operator must follow the appropriate guidance for each cryptographic algorithm. Although the cryptographic algorithms provided by the VMware's IKE Crypto Module are recommended or allowed by NIST, secure operation of these algorithms requires thorough understanding of the recommendations and appropriate limitations.
8. The VMware's IKE Crypto Module aims to be flexible and therefore it includes support for cryptographic algorithms or key lengths that were considered secure per SP 800-131A Rev. 2 and the FIPS 140-2 Implementation Guidance. It is the responsibility of the VMware's IKE Crypto Module user to ensure that disallowed algorithms, key lengths, and non-Approved services are not used.
9. Some of the implemented cryptographic algorithms offer key lengths exceeding the current NIST specifications. Such key lengths must not be used, unless following newer guidance from NIST.
   a. RSA Key Pair Generation provided by the module (FIPS 186-4 B.3.6) is only FIPS-approved for RSA modulus sizes of 2048 bits and 3072 bits. It is not permissible to generate keys using other RSA modulus sizes.

4.2 Physical Security Rules

The physical device on which the VMware's IKE Crypto Module is executed must follow the physical security rules applicable to the purpose of the device. The VMware's IKE Crypto Module is software-based and does not provide physical security.

4.3 Secure Operation Initialization Rules

The VMware's IKE Crypto Module must be linked with an application to become executable. The software code of the module (the `libsafezone-sw-fips.a` object code library or the `libsafezone-sw-fips.so` dynamically loadable library) is linked with an end application producing an executable application for the target platform. The application is installed in a platform-specific way, e.g. when purchased from an application store for the platform. In some cases there is no need for installation, e.g. when a mobile equipment vendor includes the application with the equipment.

The VMware's IKE Crypto Module is loaded by loading an application that links the library statically. The VMware's IKE Crypto Module is initialized automatically upon loading. On some platforms the module is implemented as a dynamically loadable module. In this case, the module is loaded as needed by the dynamic linker.

The VMware's IKE Crypto Module does not support operator authentication and thus does not require any authentication itself. The VMware's IKE Crypto Module is by default in FIPS-approved mode once initialized. Usually, the module does not require any special set-up or initialization except for installation.

5 Definition of SRDIs (Security Relevant Data Items) Modes of Access

This chapter specifies security relevant data items as well as the access control policy that is enforced by the VMware's IKE Crypto Module.

Each SRDI is held in the asset store accompanied by a security usage policy. The policy is set when the asset is allocated with `FL_RootKeyAllocateAndLoadValue`, `FL_AssetAllocate`, `FL_AssetAllocateBasic`, `FL_AssetAllocateSamePolicy` or `FL_AssetAllocateAndAssociateKeyExtra`. When the asset is accessed for use in a cryptographic operation, the policy is tested to ensure that the asset is eligible for the requested use. A policy typically consists of the allowed algorithm(s), the allowed strength of the algorithm, and the direction of the operation (encryption or decryption).

5.1 FIPS Approved and Allowed algorithms

The VMware's IKE Crypto Module implements the following FIPS-approved algorithms:

Table 7 – Approved, Vendor-affirmed, and CAVP Validated Cryptographic Functions

Algorithm	Implementation Details	Algorithm Certificate(s)
RSA FIPS 186-4 Signature Generation Key Pair Generation	2048, and 3072 bit keys; PKCS #1 v1.5 and PSS; SHA-224, SHA-256, SHA-384, SHA-512	RSA C460
RSA FIPS 186-4 Signature Validation	1024, 2048, and 3072 bit keys; PKCS #1 v1.5 and PSS	RSA C460
DSA FIPS 186-4 Signature Generation Domain Parameter Generation Key Pair Generation	P=2048/N=224, P=2048/N=256, P=3072/N=256; SHA-224, SHA-256, SHA-384, SHA-512	DSA C460
DSA FIPS 186-4 Signature Validation Domain Parameter Validation	P=1024/N=160, P=2048/N=224, P=2048/N=256, P=3072/N=256	DSA C460
ECDSA FIPS 186-4 Signature Generation Key Pair Generation	NIST P-224, P-256, P-384 and P-521 curves; SHA-224, SHA-256, SHA-384, SHA-512	ECDSA C460
ECDSA FIPS 186-4 Signature Validation Public Key Verification	NIST P-192, P-224, P-256, P-384 and P-521 curves	ECDSA C460
AES FIPS 197, NIST SP 800-38A	128, 192, 256 bit keys; ECB, CBC, CTR mode	AES C460
AES CCM NIST SP 800-38C	128, 192, 256 bit keys	AES C460

Algorithm	Implementation Details	Algorithm Certificate(s)
AES GCM NIST SP 800-38D	128, 192, 256 bit keys	AES C460
Triple-DES NIST SP 800-67	192 bit keys; ECB and CBC mode	Triple-DES C460
CMAC NIST SP 800-38B	128, 192, 256 bit keys	AES C460
HMAC FIPS 198-1	112-512 bit keys; SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	HMAC C460
SHS FIPS 180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512; BYTE only	SHS C460
DRBG NIST SP 800-90A	AES-128-CTR without df or reseed AES-256-CTR with df and reseed	DRBG C460, DRBG C461
KTS (KEM NIST SP 800-56B)	2048, 3072 bit keys; RSA-KEM-KWS-basic (section 9.3.3); vendor affirmed; key-wrapping; key establishment methodology provides 112 bits or 128 bits of encryption strength	N/A, Vendor-affirmed
KTS (OAEP NIST SP 800-56B)	2048, 3072 bit keys; RSA-OAEP (section 9.2.3); vendor affirmed; key-wrapping; key establishment methodology provides 112 bits or 128 bits of encryption strength	N/A, Vendor-affirmed
PBKDF NIST SP 800-132	with SHA-1, SHA-256	N/A, Vendor-affirmed
Application Specific Key Derivation Functions NIST SP 800-135rev1[1]	IKEv1 Key Derivation Functions IKEv2 Key Derivation Functions TLS 1.0/1.1 Key Derivation Functions TLS 1.2 Key Derivation Functions	CVL C460
KTS (NIST SP 800-38F Key Wrapping)	Key Wrapping function KW CIPH=AES; 128, 192, 256 bit keys Key Wrapping function KWP CIPH=AES; 128, 192, 256 bit keys	KTS (AES C460) Key establishment methodology provides between 128 and 256 bits of encryption strength
CKG (NIST SP 800-133)	Key Generation	Vendor Affirmed

The cryptographic module supports the following non-approved algorithms in the approved mode of operation as allowed:

Table 8 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Algorithm Type	Utilization
RSA Encryption (PKCS #1 v1.5)	Key Transport; 2048, 3072 bit keys	(RSA C460) Key establishment methodology provides

[1] This module supports implementation of IKEv1, IKEv2 and TLS v1.0/v1.1/v1.2 protocols. No parts of the protocols, other than KDF, have been tested by the CAVP and CMVP.

Not all algorithms/modes tested through CAVS are used within the module

		112 bits or 128 bits of encryption strength.
MD5	Message Digest; This function is only allowed as a part of an approved key transport scheme (e.g. TLS 1.0 or TLS 1.1).	
/dev/random	NDRNG	An entropy source for NIST SP 800-90A DRBG.

The VMware's IKE Crypto Module is intended for products where FIPS 140-2 approved algorithms are used.

5.2 Non-FIPS mode of operation

In the end of 2013, some of algorithms previously allowed by the NIST were disallowed. This was because 80-bits of security was considered no longer sufficient. See document NIST SP 800-131A

for details. The VMware's IKE Crypto Module implements additional key lengths for some of these algorithms (RSA, DSA, ECDSA) for compatibility with applications previously using these key sizes. These no longer approved key sizes shall only be used in non-FIPS mode of operation.

The non-FIPS validated algorithms and key sizes supported by the module are:

Table 9 – Non-Approved Cryptographic Functions for use in non-FIPS mode only

Algorithm	Implementation Details	Reason for algorithm being no longer allowed in FIPS mode.
RSA FIPS 186-2 Signature Generation	1024, 1536, 2048, 3072, and 4096 bit keys; PKCS #1 v1.5 and PSS	Transition from FIPS 186-2 to 186-4.
RSA FIPS 186-4 Signature Generation Key Pair Generation	1024 bit keys; PKCS #1 v1.5 and PSS	Key length used provides less than 112 bits of encryption strength
DSA FIPS 186-4 Signature Generation Domain Parameter Generation Key Pair Generation	P=1024/N=160	Key length used provides less than 112 bits of encryption strength
ECDSA FIPS 186-2/4 Signature Generation Key Pair Generation	NIST P-192 curve	Key length used provides less than 112 bits of encryption strength
ECDSA FIPS 186-2 Signature Generation	NIST P-224, P-256, P-384 and P-521 curves	Transition from FIPS 186-2 to 186-4.
HMAC FIPS 198-1	80-104 bit keys; SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	Key length used provides less than 112 bits of encryption strength.
KTS (KEM NIST SP 800-56B)	1024, 1536, bit keys; RSA-KEM-KWS-basic; key-wrapping	Key establishment methodology provides less than 112 bits of encryption strength

Algorithm	Implementation Details	Reason for algorithm being no longer allowed in FIPS mode.
KTS (OAEP NIST SP 800-56B)	1024, 1536 bit keys; RSA-OAEP; key-wrapping	Key establishment methodology provides less than 112 bits of encryption strength
KDF NIST SP 800-108	112-512 bit keys; SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, AES-CMAC; counter, feedback and double pipeline modes	No power-on self-test implemented
KDF NIST SP 800-108	80-104 bit keys; SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, AES-CMAC; counter, feedback and double pipeline modes	Key derivation methodology provides less than 112 bits of encryption strength.
FFC Diffie-Hellman primitive; A part of NIST SP 800-56A Rev1	Key Agreement Primitives; 1024, 2048, 3072 bit modular Diffie-Hellman groups with SHA-224, SHA-256, SHA-384 and SHA-512	KAS-FFC Component C460 SP 800-56A Rev3 transition per FIPS 140-2 IG D.8
ECC CDH primitive; A part of NIST SP 800-56A Rev1	Key Agreement Primitives; NIST P-192, P-224, P-256, P-384 and P-521 curves with SHA-224, SHA-256, SHA-384 and SHA-512	KAS-ECC CDH-Component C460 SP 800-56A Rev3 transition per FIPS 140-2 IG D.8
ECC Component A part of NIST SP800-56A Rev1	Key Agreement Primitives; NIST P-224, P-256, P-384 and P-521 curves with SHA-224, SHA-256, SHA-384 and SHA-512	KAS-ECC Component C460 SP 800-56A Rev3 transition per FIPS 140-2 IG D.8
RSA Encryption (PKCS #1 v1.5)	Key Transport; 1024, 1536 bit keys	Key establishment methodology provides less than 112 bits of encryption strength.
XTS-AES NIST SP 800-38E	256, 512 bit keys (128-bit or 256-bit encryption strength)	FIPS 140-2 IG A.9 key check not implemented
/dev/urandom	NDRNG	Disallowed by policy

5.3 Cryptographic Keys, CSPs, and SRDIs

While operating in a FIPS-compliant manner, the asset store within the VMware's IKE Crypto Module may contain the following security relevant data items (depending on which keys will be used by the user):

Table 10 – List of Cryptographic Keys, Key Components, and CSPS

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
General Keys/CSPs						

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
AES Encryption Key	AES including modes ECB, CBC, and CTR	128, 192, 256 bits	Key created for the purposes of encrypting and/or decrypting data using AES algorithm	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
AES CCM Encryption Key	AES CCM	128, 192, 256 bits	Key created for the purposes of authenticated encryption and/or decryption of data using AES and CCM algorithms	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
AES GCM Encryption Key	AES GCM	128, 192, 256 bits	Key created for the purposes of authenticated encryption and/or decryption of data using AES and GCM algorithms	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
Triple-DES Encryption Key	Triple-DES	192 bits	Key created for the purposes of encrypting and/or decrypting data using Triple-DES algorithm	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
CMAC Key	CMAC + AES	128, 192, 256 bits	Key created for the purposes of generating and verifying CMAC authentication codes	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
CMAC Verify Key	CMAC + AES	128, 192, 256 bits	Key created for the purpose of verifying CMAC authentication codes	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
KDF Key Derivation key	NIST SP 800-135 + HMAC or CMAC	112-512 bits	IKEv1/IKEv2 key derivation specified in NIST SP 800-135.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
TLS-PRF Key Derivation Key	NIST SP 800-135	112-512 bits	Key created for the purpose of key derivation using TLS1.0/TLS1.2 key derivation function presented in NIST SP 800-135.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
HMAC Key	HMAC + SHS	112-512 bits	Key created for the purposes of generating and verifying HMAC authentication codes	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
HMAC Verify Key	HMAC + SHS	112-512 bits	Key created for the purpose of verifying HMAC authentication codes	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
RSA Signing Key	RSA Private Key (CRT)	2048, 3072 bits (modulus size)	Private key for the purpose of signing data using RSA with PKCS #1v1.5 or PSS padding.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
DSA Signing Key	DSA Private Key	P=2048/N=224, P=2048/N=256, P=3072/N=256	Private key for the purpose of signing data using DSA algorithm. Includes associated domain parameters.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
ECDSA Signing Key	ECDSA Private Key	P-224, P-256, P-384, P-521	Private key for the purpose of signing data using ECDSA algorithm	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
AES Key-Wrapping Key	AES	128, 192, 256 bits	Key created for the purposes of data or key wrapping and unwrapping using NIST SP 800-38F algorithm	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
KTS (KEM) Unwrapping Key	RSA Private Key (CRT)	2048, 3072 bits	Private key for the purpose of transporting keys using RSA with KEM as specified in NIST SP 800-56B	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
KTS (OAEP) Unwrapping Key	RSA Private Key (CRT)	2048, 3072 bits	Private key for the purpose of transporting keys using RSA with OAEP as specified in NIST SP 800-56B	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
KTS (PKCS #1 v1.5) RSA Unwrapping Key	RSA Private Key (CRT)	2048, 3072 bits	Private key for the purpose of transporting keys using RSA with PKCS #1 v1.5 padding (also known as RSA Encryption)	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
Other CSPs						
DRBG CTR-128 state: Key	CTR_DRBG 128-bits with derivation function	128 bits	Key for DRBG used for random number and key/key pair generation purposes.	Entropy source	Plaintext in RAM	Power Off, FL_LibUnInit

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
DRBG CTR-128 state: V	CTR_DRBG 128-bits with derivation function	128 bits	V value for DRBG used for random number and key/key pair generation purposes.	Entropy source	Plaintext in RAM	Power Off, FL_LibUnInit
DRBG CTR-256 state: Key	CTR_DRBG 256-bits with derivation function	256 bits	Key for DRBG used for random number and key/key pair generation purposes.	Entropy source	Plaintext in RAM	Power Off, FL_LibUnInit
DRBG CTR-256 state: V	CTR_DRBG 256-bits with derivation function	128 bits	V value for DRBG used for random number and key/key pair generation purposes.	Entropy source	Plaintext in RAM	Power Off, FL_LibUnInit
Public Keys						
Software Integrity Public Key	ECDSA / Verify	NIST P-224	Public key used by Power-on Software Integrity to ensure the integrity of the Cryptographic Module.	Embedded in the software	Plaintext in persistent storage	none
RSA Verification Key	RSA Public Key	1024, 2048, 3072 bits modulus size	Public key for the purpose of verifying signed data using RSA with PKCS #1 v1.5 or PSS padding. Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
DSA Verification Key	DSA Public Key	P=1024/N=160, P=2048/N=224, P=2048/N=256, P=3072/N=256	Public key for the purpose of verifying signed data using DSA algorithm. Includes associated domain parameters. Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
ECDSA Verification Key	ECDSA Public Key	P-192, P-224, P-256, P-384, P-521	Public key for the purpose of verifying signed data using ECDSA algorithm. Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
KTS (KEM) Wrapping Key	RSA Public Key	2048, 3072 bits	Public key for the purpose of transporting keys using RSA with KEM as specified in NIST SP 800-56B. Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit
KTS (OAEP) Wrapping Key	RSA Public Key	2048, 3072 bits	Public key for the purpose of transporting keys using RSA with OAEP as specified in NIST SP 800-56B. Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit

ID	Algorithm	Size	Description	Origin	Storage	Zeroization Method
KTS (PKCS #1 v1.5) RSA Wrapping Key	RSA Public Key	2048, 3072 bits	Public key for the purpose of transporting keys using RSA with PKCS #1 v1.5 padding (also known as RSA Encryption). Not considered sensitive or CSP.	Crypto Officer, User	Plaintext in RAM	Power-off, FL_AssetFree, FL_LibUnInit

All the cryptographic keys and other security relevant materials handled by the module can be zeroized by using the cryptographic module, with the exception of the Software Integrity Public Key that is used in the self-test to validate the module.

There are three ways to zeroize a key: individual keys can be explicitly zeroized using the `FL_AssetFree` function call, all keys are zeroized once the module is uninitialized (`FL_LibUnInit`) or encounters error state, and (as all the keys handled by the module except the Software Integrity Public key are stored in RAM memory), the keys can also be zeroized by turning the power off.

5.4 Access Control Policy

The module allows controlled access to the SRDIs contained within it. The following table defines the access that an operator or an application has to each SRDI while operating the VMware's IKE Crypto Module in a given role performing a specific service (command). The permissions are categorized as a set of four separate permissions: read [R] (the SRDI can be read by this operation), write [W] (the SRDI can be written by this operation), execute [X] (the SRDI can be used in this operation), and delete [D] (the SRDI will be zeroized by this operation). If no permission is listed, then an operator outside the VMware's IKE Crypto Module has no access to the SRDI.

The operations are presented in the following tables: for secret keys, private keys, public keys, and none (operations which do not affect any of SRDI). The operations which are not appropriate for a specific key type have been omitted.

Table 11 – Secret Keys Access Policy within Services

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Secret Keys**	Security Relevant Data Item	AES Encryption Key	AES CCM Encryption Key	AES GCM Encryption Key	Triple-DES Encryption Key	CMAC Key	CMAC Verify Key	KDF Key Derivation key	TLS-PRF Key Derivation key	HMAC Key	HMAC Verify Key	AES Key-Wrapping Key	DRBG state: Key / V
Role/Service													
User role or Crypto Officer Role													
Zeroize (FL_LibUnInit)		D	D	D	D	D	D	D	D	D	D	D	D
Create Key (FL_AssetAllocate, FL_AssetAllocateBasic, FL_AssetAllocateSamePolicy, FL_AssetAllocateAndAssociateKeyExtra, FL_AssetLoadValue, FL_AssetLoadMultipart, FL_AssetLoadMultipartConvertBigInt)		W	W	W	W	W	W	W	W	W	W	W	
Copy Key (FL_AssetCopy)		W	W	W	W	W	W	W	W	W	W	W	
Delete Key (FL_AssetFree)		D	D	D	D	D	D	D	D	D	D	D	
Examine Key (FL_AssetShow, FL_AssetCheck)													
Generate Key (FL_AssetLoadRandom)		W	W	W	W	W	W	W	W	W	W	W	XW
Bulk Encryption/Decryption (FL_CipherInit, FL_CipherContinue, FL_CipherFinish)		X			X								
Authenticated Encryption/Decryption with Associated Data (FL_EncryptAuthInitRandom, FL_EncryptAuthInitDeterministic, FL_CryptAuthInit[2], FL_CryptGcmAadContinue, FL_CryptGcmAadFinish, FL_CryptAuthContinue, FL_EncryptAuthFinish, FL_EncryptAuthPacketFinish, FL_DecryptAuthFinish)			X	X									
MAC Generation (FL_MacGenerateInit, FL_MacGenerateContinue, FL_MacGenerateFinish)						X				X			
MAC Verification (FL_MacVerifyInit, FL_MacGenerateContinue, FL_MacGenerateFinish)						X	X			X	X		
DRBG Random Number Generation (FL_RbgGenerateRandom)													XW
DRBG Reseeding (FL_RbgReseed)													XW
Key Derivation (FL_KeyDeriveKdk)		W	W	W	W	W	W	XW	W	W	W	W	
TLS-PRF Key Derivation (FL_KeyDeriveKdk, FL_DeriveTlsPrf)		W	W	W	W	W	W	W	XW	W	W	W	

[2] Function may only be used to begin AES-CCM encryption operation or to continue multipacket operation with deterministic IV. In particular, the function shall not be used to initialize AES-GCM encryption.

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Secret Keys**	Security Relevant Data Item	AES Encryption Key	AES CCM Encryption Key	AES GCM Encryption Key	Triple-DES Encryption Key	CMAC Key	CMAC Verify Key	KDF Key Derivation key	TLS-PRF Key Derivation key	HMAC Key	HMAC Verify Key	AES Key-Wrapping Key	DRBG state: Key / V
Role/Service													
IKEv1 Key Derivation (FL_IkePrfExtract, FL_IKEv1ExtractSKEYID_DSA, FL_IKEv1ExtractSKEYID_PSK, FL_IKEv1ExtractSKEYID_PKE, FL_IKEv1DeriveKeyingMaterial)		W	W	W	W	W	W	XW	W	W	W	W	
IKEv2 Key Derivation (FL_IkePrfExtract, FL_IKEv2ExtractSKEYSEED, FL_IKEv2ExtractSKEYSEEDrekey, FL_IKEv2DeriveDKM)		W	W	W	W	W	W	XW	W	W	W	W	
AES Key Wrapping (FL_AssetsWrapAes, FL_AssetsWrapAes38F)		R	R	R	R	R	R	R	R	R	R	XR	
AES Key Unwrapping (FL_AssetsUnwrapAes, FL_AssetsUnwrapAes38F)		W	W	W	W	W	W	W	W	W	W	XW	
AES Data Wrapping (FL_CryptKw)												X	
AES Data Unwrapping (FL_CryptKw)												X	
Trusted Root Key Derivation (Non-Approved) (FL_TrustedKdkDerive, FL_TrustedKekdkDerive)													
Trusted KDK Key Derivation (Non-Approved) (FL_TrustedKeyDerive)		W	W	W	W	W	W	W	W	W	W	W	
Trusted Key Wrapping (Non-Approved) (FL_AssetWrapTrusted)		R	R	R	R	R	R	R	R	R	R	R	
Trusted Key Unwrapping (Non-Approved) (FL_AssetUnwrapTrusted)		W	W	W	W	W	W	W	W	W	W	W	
PBKDF2 Key Derivation (FL_KeyDerivePbkdf2)		W	W	W	W	W	W	W	W	W	W	W	
Crypto-officer Role													
Entropy Source Installation (FL_RbgInstallEntropySource, FL_RbgRequestSecurityStrength), (Non-Approved & Disallowed: FL_RbgUseNonblockingEntropySource)													W
Create Trusted Root Key (Non-Approved) (FL_RootKeyAllocateAndLoadValue)													

Table 12 – Private Keys Access Policy within Services

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Private Keys**	Security Relevant Data Item	RSA Signing Key	DSA Signing Key	ECDSA Signing Key	KTS (KEM) Unwrapping Key	KTS (OAEP) Unwrapping Key	KTS (PKCS #1 v1.5) RSA Unwrapping Key	DRBG state: Key / V
Role/Service								
User role or Crypto Officer Role								
Zeroize (FL_LibUnInit)		D	D	D	D	D	D	D
Create Key (FL_AssetAllocate, FL_AssetAllocateBasic, FL_AssetAllocateSamePolicy, FL_AssetAllocateAndAssociateKeyExtra, FL_AssetLoadValue, FL_AssetLoadMultipart, FL_AssetLoadMultipartConvertBigInt)		W	W	W	W	W	W	
Copy Key (FL_AssetCopyValue)		W	W	W	W	W	W	
Delete Key (FL_AssetFree)		D	D	D	D	D	D	
Examine Key (FL_AssetShow, FL_AssetCheck)								
Generate Key (FL_AssetLoadRandom)								XW
Generate Key Pair (FL_AssetGenerateKeyPair)		W	W	W	W	W	W	XW
DSA Domain Parameter and Key Pair Generation (FL_AssetGenerateKeyPair)			W					XW
Signature Generation (FL_HashSignFips186, FL_HashSignPkcs1, FL_HashSignPkcs1Pss)		X	X	X				XW
AES Key Wrapping (FL_AssetsWrapAes, FL_AssetsWrapAes38F)		R	R	R	R	R	R	
AES Key Unwrapping (FL_AssetsUnwrapAes, FL_AssetsUnwrapAes38F)		W	W	W	W	W	W	
Trusted Key Wrapping (Non-Approved) (FL_AssetWrapTrusted)		R	R	R	R	R	R	
Trusted Key Unwrapping (Non-Approved) (FL_AssetUnwrapTrusted)		W	W	W	W	W	W	
PBKDF2 Key Derivation (FL_KeyDerivePbkdf2)		W	W	W	W	W	W	
Diffie-Hellman Key Agreement (Non-Approved) (FL_DeriveDh)								
Elliptic Curve Diffie-Hellman Key Agreement (Non-Approved) (FL_DeriveDh)								

Table 13 – Public Keys Access Policy within Services

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Public Keys**	Security Relevant Data Item	Software Integrity Public Key	RSA Verification Key	DSA Verification Key	ECDSA Verification Key	KTS (KEM) Wrapping Key	KTS (OAEP) Wrapping Key	KTS (PKCS #1 v1.5) RSA Wrapping Key	DRBG state: Key / V
Role/Service									
User role or Crypto-Officer Role									
Zeroize (FL_LibUnInit)			D	D	D	D	D	D	D
On-demand self-test (FL_LibSelfTest)		X							
Create Key (FL_AssetAllocate, FL_AssetAllocateBasic, FL_AssetAllocateSamePolicy, FL_AssetAllocateAndAssociateKeyExtra, FL_AssetLoadValue, FL_AssetLoadMultipart, FL_AssetLoadMultipartConvertBigInt)			W	W	W	W	W	W	
Copy Key (FL_AssetCopyValue)			W	W	W	W	W	W	
Delete Key (FL_AssetFree)			D	D	D	D	D	D	
Examine Key (FL_AssetShow, FL_AssetCheck)			RX	RX	RX	RX	RX	RX	
Generate Key Pair (FL_AssetGenerateKeyPair)			W	W	W	W	W	W	XW
DSA Domain Parameter and Key Pair Generation (FL_AssetGenerateKeyPair)				W					XW
Public Key Validation (FL_AssetCheck)			X	X	X	X	X	X	
DSA Domain Parameter Verification (FL_AssetCheck)				X					
Signature Verification (FL_HashVerifyFips186, FL_HashVerifyPkcs1, FL_HashVerifyRecoverPkcs1, FL_HashVerifyPkcs1Pss)			X	X	X				
Diffie-Hellman Key Agreement (Non-Approved) (FL_DeriveDh)									
Elliptic Curve Diffie-Hellman Key Agreement (Non-Approved) (FL_DeriveDh)									
Crypto-officer Role									
Module Initialization (FL_LibInit) (This function is automatically invoked upon loading the module)		X							XW

Table 14 – Services not using SRDI

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Services not using any SRDI**
Role/Service
User role or Crypto Officer Role
Show Status (FL_LibStatus)
Digest Generation (FL_HashInit, FL_HashContinue, FL_HashFinish, FL_HashFinishKeep, FL_HashSingle)

Table 15 – Non-FIPS 140-2 Security Relevant Services

VMware's IKE Crypto Module SRDI/Role/Service Access Policy **Non-FIPS 140-2 Security Relevant Services**
Role/Service
Services provided for convenience which offer no FIPS 140-2 Security Relevant Functions
Show Version (FL_LibVersion)
Test DRBG (FL_RbgTestVector)
Check Free Space in Key Store (FL_AssetStoreStatus)

5.5 User Guide

Some of the FIPS Publications or NIST Special Publications require that the Cryptographic Module Security Policy mentions important configuration items for those algorithms. The user of the module shall observe these rules.

5.5.1 NIST SP 800-56A Rev. 1: Key Agreement Primitives

The module supports KAS-FFC and KAS-ECC primitives compliant with SP 800-56A Rev. 1. Since SP 800-56A Rev. 3 is now required per FIPS 140-2 IG D.8, they are not Approved for usage in the Approved mode of operation.

5.5.2 NIST SP 800-108: Key Derivation Functions

All three key derivation functions, Counter Mode, Feedback Mode and Double-Pipeline Iteration Mode are supported. Since no power-on self-tests are implemented for these key derivation functions, they are not Approved for usage in the Approved mode of operation.

5.5.3 NIST SP 800-132: Password-Based Key Derivation Function

The key derived using NIST SP 800-132 shall only be used for storage purposes.

Both options presented in NIST SP 800-132 for deriving the Data Protection Key from the Master Key are supported.

The VMware's IKE Crypto Module does not limit the length of the passphrase used in NIST SP 800-132 PBKDF key derivation. The upper bound for the strength of passwords usually used is between 5 or 6 bits per character. Thus, for security over 64 bits, the passwords must generally be longer than 12 characters.

Minimum requirements and limits for NIST SP 800-132:

- There is no maximum for length of salt used, but at least 128 bits (16 bytes) of salt value must be randomly generated.
- Iteration count shall be as large as possible. Iteration count used must be at least 1000 to meet minimum requirements of NIST SP 800-132. However, often it is recommendable to use much larger iteration counts, such as 100000 or 1000000, when user-perceived performance is not critical.
- Resulting MK (Master key) can be used directly as the Data protection key, or as input to KDF or as a decryption key for Encrypted Data protection key.

5.5.4 NIST SP 800-38D: Galois/Counter Mode

The FIPS 140-2 Implementation Guidance A.5 applies to AES-GCM usage with this module.

Item 1 in IG A.5 forbids using external IV for encryption via the `FL_CryptAuthInit` function. However, the `FL_CryptAuthInit` function is still used for decryption and the `FL_CryptAuthInit` function is used for subsequent encryption operations for operation sequences started with the `FL_EncryptAuthInitDeterministic` function.

The operator must use the `FL_EncryptAuthInitRandom` function if random IV generation (IG A.5 item 2) is required, or in case of deterministic IV generation (IG A.5 item 3), the `FL_EncryptAuthInitDeterministic` function.

Note: If IV is generated internally in a deterministic manner, then FIPS 140-2 Implementation Guidance A.5: Item B3 applies: In case a module’s power is lost and then restored, the key used for the AES GCM encryption/decryption must be re-distributed.

5.5.5 NIST SP 800-90A: Deterministic Random Bit Generator

The module generates cryptographic keys whose strengths are modified by available entropy. No assurance of the minimum strength of the generated keys is given by the module. Depending on the platform, the module provides access to different entropy sources.

By default, the VMware's IKE Crypto Module DRBG uses /dev/random as the entropy source on platforms that provide such an entropy device. This entropy generation path is merely a convenience default. The quality of entropy coming from /dev/random is not measured by the VMware's IKE Crypto Module.

It is possible, but disallowed by policy, to use the function FL_RbgUseNonblockingEntropySource to configure /dev/urandom as the entropy source. When using /dev/urandom as the entropy source, the module assumes the quality of the entropy source to be 128 bits. The difference between /dev/random and /dev/urandom is that when the entropy source does not know if there is sufficient entropy available, /dev/random will block and /dev/urandom will generate pseudo-random values based on available entropy. The quality of entropy coming from /dev/urandom is not measured by the VMware's IKE Crypto Module.

If Crypto Officer uses /dev/random or /dev/urandom as entropy source, it is up to Crypto Officer to configure it suitably to provide reasonable security. Crypto Officer can also provide an entropy function which overrides the default entropy source.

5.5.6 NIST SP 800-67 Rev 2: Triple-DES Encryption

The module support Triple-DES encryption algorithm. The algorithm has tight restrictions for maximum number of encryptions with a single key. It is allowed to perform at most 2^20 (IETF protocols) or 2^16 (non-IETF protocols) data block encryptions with the same Triple-DES key.

The module does not enforce these limits, instead the user of the module is responsible for ensuring their use of the module meets these restrictions.

According to NIST SP 800-131A Rev 2, Triple-DES Encryption is deprecated algorithm and is becoming disallowed after 2023. It is recommended that the users move to AES algorithm for symmetric encryption needs.

5.5.7 NIST SP 800-133: Key Generation

The module allows key generation. Key generation will use one of the random number generators, NIST SP 800-90A DRBG-CTR AES-256 or DRBG-CTR AES-128. AES-256 based DRBG is used to generate symmetric keys and many of asymmetric keys. AES-128 DRBG is used in generation of some asymmetric keys of up-to 128 bit equivalent security strength (such as RSA-2048 and RSA-3072 keys).

The output of the approved DRBG is used unmodified when symmetric keys are generated. It is also used unmodified as random input for asymmetric key generation.

6 Self-Tests

6.1 Power-Up Self-Tests

The VMware's IKE Crypto Module includes the following power-up self-tests:

- Software Integrity Test (using ECDSA Verify with NIST P-224)
- KAT test for SHA-1
- KAT test for SHA-512
- KAT test for HMAC SHA-256
- KAT test for AES encryption (CBC, 128-bit key)
- KAT test for AES decryption (CBC, 128-bit key)
- KAT test for AES encryption (CCM, 128-bit key)
- KAT test for AES decryption (CCM, 128-bit key)
- KAT test for AES encryption (GCM, 128-bit key)
- KAT test for AES decryption (GCM, 128-bit key)
- KAT test for AES encryption (XTS, 128-bit key strength)
- KAT test for AES decryption (XTS, 128-bit key strength)
- KAT test for CMAC, 192-bit key
- KAT test for Triple-DES encryption (CBC, 192-bit key)
- KAT test for Triple-DES decryption (CBC, 192-bit key)
- KAT for RSA 2048-bit (PKCS #1 v1.5)
- KAT for DSA (signing P=2048/N=256; verification P=1024/N=160)
- KAT for ECDSA Signing (NIST P-224)
- KAT for KTS: RSA Key Wrapping 2048-bit (RSA-OAEP)
- KAT for Diffie-Hellman
- KAT for EC Diffie-Hellman
- AES-CTR-256 DRBG self-test

The self-tests are invoked automatically upon loading the VMware's IKE Crypto Module. The initialization function FL_LibInit is executed via DEP (default entry point) as specified in FIPS 140-2 Implementation Guidance 9.10.

Any error during the power-up self-tests will result in a module transition to the error state. There are two possible ways to recover from the error state:

- Reinitializing the module with the API function sequences FL_LibUnInit and FL_LibInit.
- Power-cycling the device and reinitialize the module with the API function FL_LibInit.

The FL_LibStatus API function can be used to obtain the module status. It returns FL_STATUS_INIT when the module has not yet been initialized and FL_STATUS_ERROR when the module is in error state.

As it is recommended to self-test cryptographic components (like DRBG) frequently, the module

provides the capability to invoke the self-tests manually (on demand) with the FL_LibSelfTest API function. The important difference between the manually invoked self-tests and the automatically invoked self-tests when initializing the module is that the manually invoked self-tests will not cause zeroization of the key material currently loaded in the module, providing the tests execute successfully.

In general, if a self-test fails, the module will transition to the error state and the return value (status) of the invoked API function will be something other than `FLR_OK`, depending on the current situation.

6.2 Conditional Self tests

The VMware's IKE Crypto Module contains the following conditional self-tests:

- Pair-wise consistency check for key pairs created for digital signature purposes (DSA, FIPS 186-4)
- Pair-wise consistency check for key pairs created for digital signature purposes (RSA, FIPS 186-4)
- Pair-wise consistency check for key pairs created for digital signature purposes (ECDSA, FIPS 186-4)
- Continuous random number generator test for Approved DRBG.
- Continuous random number generator test for non-Approved RBG `/dev/random`.
- Continuous random number generator test for non-Approved (disallowed) RBG `/dev/urandom`.

The conditional self-tests for manual key entry and software/firmware load or bypass are not provided, as these are not applicable.

Any error during the conditional self-tests will result in a module transition to the error state. The ways to recover from the error state are listed in section 6.1.

7 Mitigation of Other Attacks

The module contains an implementation of the RSA algorithm with data independent processing time for signing and decryption operations. This makes it harder to attack the RSA implementation via timing attacks.

The module does not mitigate other attacks outside the scope of FIPS 140-2.

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