ORACLE®
Communications

FIPS 140-2 Non-Proprietary Security Policy

# Acme Packet VME

FIPS 140-2 Level 1 Validation

Software Versions: S-Cz8.2.0 and S-Cz8.2.0p5

Date: November 27th, 2019

**Title:** Acme Packet VME Non-Proprietary Security Policy
**Date:** November 27th, 2019
**Author:** Acumen Security, LLC.
**Contributing Authors:**
Oracle Communications Engineering
Oracle Security Evaluations – Global Product Security

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment



**Hardware and Software, Engineered to Work Together**

Table of Contents

## List of Tables



## List of Figures

# 1. Introduction

## 1.1 Overview

This document is the Security Policy for the Acme Packet VME developed by Oracle Communications. Acme Packet VME is also referred to as "the module" or "module". This Security Policy specifies the security rules under which the module shall operate to meet the requirements of FIPS 140-2 Level 1. It also describes how the Acme Packet VME functions to meet the FIPS requirements, and the actions that operators must take to maintain the security of the module.

This Security Policy describes the features and design of the Acme Packet VME module using the terminology contained in the FIPS 140-2 specification. FIPS 140-2, Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-2. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

## 1.2 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. The Submission Package contains:

- Oracle Non-Proprietary Security Policy
- Oracle Vendor Evidence document
- Finite State Machine
- Entropy Assessment Document
- Other supporting documentation as additional references

With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Oracle and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Oracle.

# 2. Acme Packet VME

## 2.1 Functional Overview

The Acme Packet VME is specifically designed to meet the unique price performance and manageability requirements of the small to medium sized enterprise and remote office/ branch office. Ideal for small site border control and Session Initiation Protocol (SIP) trunking service termination applications, the Acme Packet VME deliver Oracle's industry leading ESBC capabilities in binary packaged executable that can be run in a virtual environment.

Acme Packet VME addresses the unique connectivity, security, and control challenges enterprises often encounter when extending real-time voice, video, and UC sessions to smaller sites. The appliance also helps enterprises contain voice transport costs and overcome the unique regulatory compliance challenges associated with IP telephony. An embedded browser based graphical user interface (GUI) simplifies setup and administration.

# 3. Cryptographic Module Specification

## 3.1 Definition of the Cryptographic Module

The logical cryptographic boundary of the module consists of the Oracle VME ISO image called "nnSCZ820-img.iso " for version S-Cz8.2.0 and "nnSCZ820p5-img.iso" for version S-Cz8.2.0p5.

Figure 1 shows the logical block diagram (red-dotted line) of the module executing in memory and its interactions with the hypervisor through the module's defined logical cryptographic boundary. The module interacts directly with the hypervisor, which runs directly on the host system.

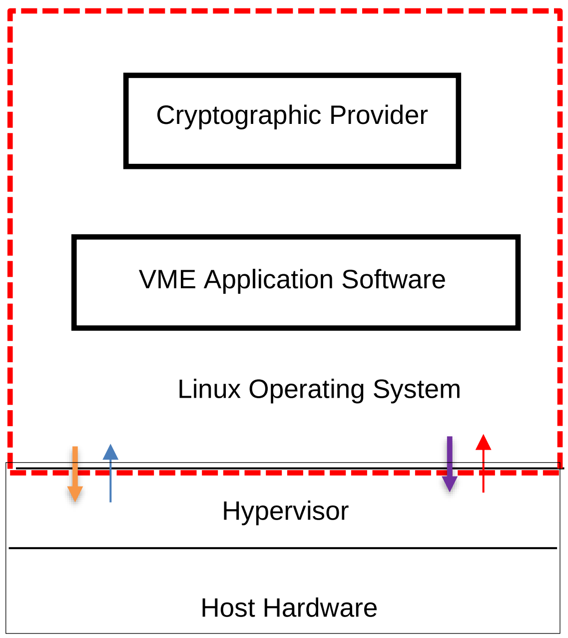
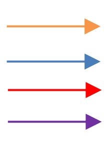



**Figure 1: VME Logical Cryptographic Boundary**




## 3.2 Definition of the Physical Cryptographic Boundary

The module consists of binary packaged into an executable that can be run in a virtual environment. The module is classified as a multi-chip standalone cryptographic module. The physical cryptographic boundary is defined as the hard enclosure of the host system on which it runs and no components are excluded from the requirements of FIPS PUB 140-2.

## 3.3 FIPS 140-2 Validation Scope

The Acme Packet VME appliances are being validated to overall FIPS 140-2 Level 1 requirements. See Table 1 below.

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles and Services and Authentication | 2 |
| Finite State Machine Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 1 |
| Self-Tests | 1 |
| Design Assurance | 3 |
| Mitigation of Other Attacks | N/A |

**Table 1: FIPS 140-2 Security Requirements**

## 3.4 Approved or Allowed Security Functions

The Acme Packet VME contains the following FIPS Approved Algorithms listed in Table 2 (Oracle Acme Packet Cryptographic Library Acme Packet Virtual Machine Edition (VME)) and Table 3 (Oracle Acme Packet Mocana Cryptographic Library Acme Packet Virtual Machine Edition (VME)):

| | Approved or Allowed Security Functions | Certificate |
|---|---|---|
| ***Symmetric Algorithms*** | | |
| AES | CBC, ECB, CTR, GCM; Encrypt/Decrypt; Key Size = 128, 256 | C 144 |
| Triple DES[1] | CBC; Encrypt/Decrypt; Key Size = 192 | C 144 |
| ***Secure Hash Standard (SHS)*** | | |
| SHS | SHA-1, SHA-256, SHA-384, SHA-512 | C 144 |
| ***Data Authentication Code*** | | |
| HMAC | HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 | C 144 |
| ***Asymmetric Algorithms*** | | |

[1] Triple-DES was CAVP tested but is not utilized by the services associated with the Oracle Acme Packet Cryptographic Library.

| | | |
|---|---|---|
| RSA | RSA: FIPS186-4:<br>186-4 KEY(gen): FIPS186-4_Random_e<br>ALG[ANSIX9.31] SIG(gen) (2048 SHA(1, 256 , 384))<br>ALG[ANSIX9.31] SIG(Ver) (2048 SHA(1, 256, 384))<br><br>RSA: FIPS186-2 :<br>ALG[ANSIX9.31] SIG(gen) (4096 SHA (256,384))<br>ALG[ANSIX9.31] SIG(Ver) (2048 SHA(1, 256, 384)), (4096 SHA (1, 256, 384))<br><br>RSA: FIPS186-4:<br>186-4 KEY(gen):<br>FIPS186-4_Random_e ALG[ANSIX9.31] SIG(gen) (2048 SHA(1, 256 , 384), (4096 SHA (256,384))<br>SIG(Ver) (2048 SHA(1, 256, 384))<br><br>RSA: FIPS186-2<br>Signature Verification 9.31:<br>Modulus lengths: 2048, 4096<br>SHAs: SHA-1, SHA-256, SHA-384 | C 144 |
| ECDSA | Firmware: FIPS186-4<br>PKG: CURVES (P-256, P-384 Testing Candidates)<br>SigGen: CURVES (P-256: (SHA-256, 384) P-384: (SHA-256, 384)<br>SigVer: CURVES (P-256: (SHA-256, 384) P-384: (SHA-256, 384)) | C 144 |
| ***Random Number Generation*** | | |
| DRBG | CTR_DRBG: [ Prediction Resistance Tested: Not Enabled; BlockCipher_Use_df: (AES-256)]<br>Hash_Based DRBG: [ Prediction Resistance Tested: Not Enabled (SHA-1) | C 144 |
| ***Key establishment*** | | |
| Key Derivation | SNMP KDF, SRTP KDF, TLS KDF (TLS Version: v1.0/1.1, v1.2) | C 144 |
| ***Key Transport*** | | |
| KTS | KTS (AES Cert. # C144 and HMAC Cert. # C144; key establishment methodology provides 128 or 256 bits of encryption strength); | |

**Table 2: Approved and Allowed Security Functions Acme Packet Cryptographic Library Virtual Machine Edition (VME)**

| | Approved or Allowed Security Functions | Certificate |
|---|---|---|
| ***Symmetric Algorithms*** | | |
| AES | CBC; Encrypt/Decrypt; Key Size = 128, 256 | C 142 |
| Triple DES[2] | CBC; Encrypt/Decrypt; Key Size = 192 | C 142 |
| ***Secure Hash Standard (SHS)*** | | |
| SHS | SHA-1, SHA-256, SHA-384, SHA-512 | C 142 |
| ***Data Authentication Code*** | | |
| HMAC | HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 | C 142 |

[2] Per IG A.13 the same Triple-DES key shall not be used to encrypt more than 2^20 64-bit blocks of data.

| *Asymmetric Algorithms* | | |
|---|---|---|
| RSA | RSA: 186-4:<br>186-4 KEY(gen): FIPS186-4_Random_e PKCS1.5: SIG(Ver) (1024 SHA(1); (2048 SHA (1)) | C 142 |
| ***Key Establishment*** | | |
| Key Derivation | SSH KDF, IKEv1/IKEv2 KDF | C 142 |
| ***Key Transport*** | | |
| KTS | KTS (AES Cert. # C142 and HMAC Cert. # C142; key establishment methodology provides 128 or 256 bits of encryption strength); | |

**Table 3: Approved and Allowed Security Functions Oracle Acme Packet Mocana Cryptographic Library Virtual Machine Edition (VME)**

**Note:** P-384 for ECDSA was CAVP tested but is not utilized by the module's services.

## 3.5 Non-Approved But Allowed Security Functions

The following are considered non-Approved but allowed security functions:

| Algorithm | Usage |
|---|---|
| EC-Diffie-Hellman | CVL Certs. #C:144 and #C:142, key agreement, key establishment methodology provides 128 or 192-bits of encryption strength. |
| Diffie-Hellman | CVL Certs. #C:144 and #C:142, key agreement, key establishment methodology provides 112-bits of encryption strength. |
| RSA Key Wrapping | Key wrapping, key establishment methodology provides 112-bits of encryption strength. |
| NDRNG | Used for seeding the NIST SP 800-90A Hash_DRBG and CTR_DRBG. Per FIPS 140-2 IG 7.14 scenario 1 (a).<br><br>The module provides a minimum of 440 bits of entropy input for the Hash_DRBG. The input length for the CTR_DRBG depends on the size of the AES key used. If the AES key length is 128 bits, the seed size is 256 bits. If the AES key length is 256 bits, then the seed size is 384 bits. |
| MD5 (TLS 1.0/1.1/1.2) | MACing: HMAC MD5, Hashing: MD5 |

**Table 4: Non-Approved but Allowed Security Functions**

## 3.6 Non-Approved Security Functions and Services

The following services are considered non-Approved and may not be used in a FIPS-approved mode of operation:

| Service | Non-Approved Security Functions |
|---|---|
| SSH | Asymmetric Algorithms: DSA, Symmetric Algorithms: Rijndael, AES GCM, 192-Bit AES CTR |
| SNMP | Hashing: MD5, Symmetric Algorithms: DES |
| SRTP | Hashing: MD5 |
| IKEv1, IKEv2 | Hashing: MD5, Symmetric Algorithms: 192-Bit AES CBC |
| TLS 1.0/1.1/1.2 | Symmetric Algorithms: DES |

| | |
|---|---|
| Diffie-Hellman | Key agreement, less than 112 bits of encryption strength. |
| RSA Key Wrapping | Key wrapping, less than 112 bits of encryption strength. |

**Table 5: Non-Approved Disallowed Functions**

Services listed in the previous table make use of non-compliant cryptographic algorithms. Use of these algorithms is prohibited in a FIPS-approved mode of operation. Some of these services may be allowed in FIPS mode when using allowed algorithms (as specified in section 8.1)

## 3.7 Vendor Affirmed Security Functions

The following services are considered non-Approved and may not be used in a FIPS-approved mode of operation:

| Algorithm | Vendor Affirmed Security Functions |
|---|---|
| CKG | In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133 (vendor affirmed). The resulting generated symmetric keys and the seed used in the asymmetric key generation are the unmodified output from an NIST SP 800-90A DRBG. |

**Table 6: Vendor Affirmed Functions**

# 4. Module Ports and Interfaces

Oracle Virtual Machine edition is a virtualized cryptographic module that meets the overall Level 1 FIPS 140-2 requirements. The module interfaces can be categorized as follows:

- Data Input Interface
- Data Output Interface
- Control Input interface
- Status Output Interface
- Power Interface

The table below provides a mapping of ports for the Oracle VME:

| FIPS 140 Interface | Physical Port | VM Port | Logical Interface | Information Input/Output |
|---|---|---|---|---|
| Data Input | Host System Ethernet (10/100/1000) Ports, Host System USB Ports. | • Virtual Ethernet Ports,<br>• Virtual USB Ports. | API Input Data and Parameters. | Cipher text<br><br>Plain text |
| Data Output | Host System Ethernet (10/100/1000) Ports, Host System USB Ports. | • Virtual Ethernet Ports,<br>• Virtual USB Ports. | API Output Data and Parameters. | Cipher text<br><br>Plain Text |
| Control Input | Host System Ethernet (10/100/1000) Ports, Host System Serial Ports. | • Virtual Ethernet Ports,<br>• Virtual Serial Ports. | API Command Input Parameters. | • Plaintext control input via console port (configuration commands, operator passwords)<br>• Ciphertext control input via network management (EMS control, CDR accounting, CLI management) |
| Status Output | Host System Ethernet (10/100/1000) Ports, Host System Serial Ports. | • Virtual Ethernet Ports,<br>• Virtual Serial Ports. | API Status Output Parameters. | Plaintext Status Output via Console Port.<br><br>Ciphertext Status Output via network management. |
| Power | Host Power Plug | NA | N/A | N/A |

**Table 7: Mapping of FIPS 140 Logical interfaces to Logical Ports**

## 5. Physical Security

The module is comprised of software only and thus does not claim any physical security.

## 6. Roles and Services

As required by FIPS 140-2 Level 1, there are three roles (a Crypto Officer Role, User Role, and Unauthenticated Role) in the module that operators may assume. The module supports role-based authentication, and the respective services for each role are described in the following sections. The below table gives a high-level description of all services provided by the module and lists the roles allowed to invoke each service.

| Operator Role | Summary of Services |
| --- | --- |
| User | • View configuration versions and system performance data<br>• Test pattern rules, local policies, and session translations<br>• Display system alarms. |
| Crypto-Officer | Allowed access to all system commands and configuration privileges |
| Unauthenticated | • Request Authentication<br>• Show Status<br>• Initiate self-tests |

**Table 8: Service Summary**

### 6.1 Operator Services and Descriptions

The below table provides a full description of all services provided by the module and lists the roles allowed to invoke each service.

| U | CO | Service Name | Service Description | Keys and CSP(s) | Access Type(s) |
| --- | --- | --- | --- | --- | --- |
| | X | Configure | Initializes the module for FIPS mode of operation | HMAC-SHA-256 key | R, W, X |
| | X | Zeroize CSP's | Clears keys/CSPs from memory and disk | All CSP's | Z |
| | X | Software Update | Updates software | Software Integrity Key (RSA) | R, X |
| | X | Bypass | Configure bypass using TCP or UDP and viewing bypass service status | HMAC-SHA-256 Bypass Key | R, W, X |

| U | CO | Service Name | Service Description | Keys and CSP(s) | Access Type(s) |
|---|---|---|---|---|---|
| X | X | Decrypt | Decrypts a block of data Using AES or Triple-DES in FIPS Mode | TLS Session Keys (AES128)<br>TLS Session Keys (AES256)<br>SSH Session Key (AES128)<br>SSH Session Key (AES256)<br>SRTP Session Key (AES-128)<br>SNMP Privacy Key (AES-128)<br>IKE Session Encryption Key (Triple-DES, AES-128, AES-256)<br>IPsec Session Encryption Key (Triple-DES, AES-128 or AES-256) | X<br>X<br>X<br>X<br>X<br>X<br>X<br>X |
| X | X | Encrypt | Encrypts a block of data Using AES or Triple-DES, in FIPS Mode | TLS Session Keys (AES128)<br>TLS Session Keys (AES256)<br>SSH Session Key (AES128)<br>SSH Session Key (AES256)<br>SRTP Session Key (AES-128)<br>SNMP Privacy Key (AES-128)<br>IKE Session Encryption Key (Triple-DES, AES-128, AES-256)<br>IPsec Session Encryption Key (Triple-DES, AES-128 or AES-256) | X<br>X<br>X<br>X<br>X<br>X<br>X<br>X |
| X | X | Generate Keys | Generates AES or Triple-DES for encrypt/decrypt operations. | TLS Session Keys (AES128)<br>TLS Session Keys (AES256)<br>SSH Session Key (AES128)<br>SSH Session Key (AES256)<br>SRTP Session Key (AES-128)<br>SNMP Privacy Key (AES-128)<br>IKE Session Encryption Key (Triple-DES, AES-128, AES-256)<br>IPsec Session Encryption Key (Triple-DES, AES-128 or AES-256) | R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W |

| U | CO | Service Name | Service Description | Keys and CSP(s) | Access Type(s) |
|---|---|---|---|---|---|
| | | | Generates Diffie-Hellman, EC Diffie-Hellman, and RSA keys for key transport/key establishment. | Diffie-Hellman Public Key (DH)<br>Diffie-Hellman Private Key (DH)<br>EC Diffie-Hellman Public Key (ECDH)<br>EC Diffie-Hellman Private Key (ECDH)<br>SSH authentication private Key (RSA)<br>SSH authentication public key (RSA)<br>TLS authentication private Key (ECDSA/RSA)<br>TLS authentication public key (ECDSA/RSA)<br>TLS premaster secret,<br>TLS Master secret,<br>SRTP Master key<br>IKE Private Key (RSA)<br>IKE Public Key (RSA)<br>SKEYSEED<br>SKEYID<br>SKEYID_d | R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W<br>R, W |
| X | X | Verify | Used as part of the TLS, SSH protocol negotiation | SSH authentication private Key (RSA)<br>SSH authentication public key (RSA)<br>TLS authentication private Key (ECDSA/RSA)<br>TLS authentication public key (ECDSA/RSA)<br>Diffie-Hellman Public Key (DH)<br>Diffie-Hellman Private Key (DH)<br>EC Diffie-Hellman Public Key (ECDH)<br>EC Diffie-Hellman Private Key (ECDH) | X<br>X<br>X<br>X<br>X<br>X<br>X<br>X |
| X | X | Generate Seed | Generate an entropy_input for Hash_DRBG, CTR DRBG | DRBG Seed<br>DRBG Entropy Input String | R, W, X |
| X | X | Generate Random Number | Generate random number. | DRBG C<br>DRBG V<br>DRBG Key | R, W, X<br>R, W, X<br>R, W, X |
| X | X | HMAC | Generate HMAC | SNMP Authentication Key<br>SRTP Authentication Key<br>SSH Integrity Keys<br>TLS Integrity Keys<br>IPsec Session Authentication Key<br>IKE Session Authentication Key | X<br>X<br>X<br>X<br>X<br>X |
| X | X | Generate Certificate | Generate certificate | Web UI Certificate | R, W, X |

| U | CO | Service Name | Service Description | Keys and CSP(s) | Access Type(s) |
|---|---|---|---|---|---|
| X | X | Authenticate | Authenticate Users | Operator Password | R, W, X |

**R – Read, W – Write, X – Execute, Z - Zeroize**

**Table 9: Operator Services and Descriptions**

Note: TLS, SRTP and SNMP protocols use the Oracle Acme Packet Cryptographic library.
Note: SSH, IKEv2 and IPSec use the Oracle Acme Packet Mocana Cryptographic library.

## 6.2 Unauthenticated Services and Descriptions

The below table provides a full description of the unauthenticated services provided by the module:

| Service Name | Service Description |
|---|---|
| On-Demand Self-Test Initialization | This service initiates the FIPS self-test when requested. |
| Show Status | This service shows the operational status of the module |
| Factory Reset Service | Factory Reset Service - This service restores the module to factory defaults |

**Table 10: Operator Services and Descriptions**

## 6.3 Operator Authentication

### 6.3.1 Crypto-Officer: Password-Based Authentication

In FIPS-approved mode of operation, the module is accessed via Command Line Interface over the Console ports or via SSH, SNMPv3 or HTTPS over the Network Management Ports. Other than status functions available by viewing the Status LEDs, the services described are available only to authenticated operators.

| Method | Probability of a Single Successful Random Attempt | Probability of a Successful Attempt within a Minute |
|---|---|---|
| Password-Based (CO and User Authentication) | Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. The probability of a successful random attempt is 1/94^8, which is less than 1/1,000,000. | Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/94^8, which is less than 1/100,000. |

| Method | Probability of a Single Successful Random Attempt | Probability of a Successful Attempt within a Minute |
|---|---|---|
| SNMPv3 Passwords | Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. The probability of a successful random attempt is 1/94^8, which is less than 1/1,000,000. | Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/94^8, which is less than 1/100,000. |
| Password-Based (Challenge Response) | Passwords must be a minimum of 12 numeric characters. 0-9, yielding 10 choices per character. The probability of a successful random attempt is 1/10^12, which is less than 1/1,000,000. | Passwords must be a minimum of 12 numeric characters. 0-9, yielding 10 choices per character. Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/10^12, which is less than 1/100,000. |

**Table 11: Crypto-Officer and User Authentication**

### 6.3.2 User: Password-Based Authentication

The module also supports authentication via digital certificates for the User Role as implemented by the TLS and SSH protocols. The module supports a public key-based authentication with 2048-bit RSA and 2048-bit ECDSA keys.

| Method | Probability of a Single Successful Random Attempt | Probability of a Successful Attempt within a Minute |
|---|---|---|
| Certificate-Based | A 2048-bit RSA/ECDSA key has at least 112-bits of equivalent strength. The probability of a successful random attempt is 1 /2^112, which is less than 1/1,000,000. | Assuming the module can support 60 authentication attempts in one minute, the probability of a success with multiple consecutive attempts in a one-minute period is 60/2^112, which is less than 1/100,000. |

**Table 12: User Authentication**

## 6.4 Key and CSP Management

The following keys, cryptographic key components and other critical security parameters are contained in the module. No parts of the SSH, TLS, IKEv1/IKEv2, SNMP or SRTP protocols, other than the KDF, have been tested by the CAVP and CMVP.

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| Operator Passwords | Generated by the crypto officer as per the module | **Agreement**: NA | Virtual Hard Disk | Authentication of the crypto officer and user |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | policy | **Entry**: Manual entry via console or SSH management session<br><br>**Output**: Output as part of HA direct physical connection to another box | | |
| Software Integrity Key (RSA) | Generated externally | **Entry**: RSA (2048 bits) entered as part of software image<br><br>**Output**: Output as part of HA direct physical connection to another box | Virtual Hard Disk | Public key used to verify the integrity of software and updates |
| DRBG Entropy Input String | Generated internally from hardware sources | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used in the random bit generation process |
| DRBG Seed | Generated internally from hardware sources | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Entropy used in the random bit generation process |
| DRBG C | Internal value used as part of SP 800-90a HASH_DRBG | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used in the random bit generation process |
| DRBG V | Internal value used as part of SP 800-90a HASH_DRBG | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used in the random bit generation process |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| DRBG V | Internal value used as part of SP 800-90a CTR_DRBG | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used in the random bit generation process |
| DRBG Key | Internal value used as part of SP 800-90a CTR_DRBG | **Agreement**: NA<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used in the random bit generation process |
| Diffie-Hellman Public Key (DH) 2048-bit | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: Diffie-Hellman<br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used to derive the secret session key during DH key agreement protocol |
| Diffie-Hellman Private Key (DH) 224-bit | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: Diffie-Hellman<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used to derive the secret session key during DH key agreement protocol |
| ECDH Public Key (P-256) | Internal generation by FIPS-approved CTR_DRBG | **Agreement:** EC Diffie-Hellman.<br><br>**Entry: NA**<br><br>**Output: None** | Volatile RAM | Used to derive the secret session key during ECDH key agreement protocol |
| ECDH Private Key (P-256) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: EC Diffie-Hellman.<br><br>**Entry**: NA<br><br>**Output**: None | Volatile RAM | Used to derive the secret session key during ECDH key agreement protocol |
| SNMP Privacy Key (AES-128) | NIST SP 800-135 KDF | **Agreement**: NIST SP 800-135 KDF | Volatile RAM | For encryption / decryption of SNMP session traffic |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | | **Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | | |
| SNMP Authentication Key (HMAC-SHA512) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | 160-bit HMAC-SHA-512 for message authentication and verification in SNMP |
| SRTP Master Key (AES-128) | Internal generation by FIPS-approved Hash_DRBG | **Agreement**: Diffie-Hellman<br><br>**Entry**: NA<br><br>**Output**: encrypted or output as part of HA direct physical connection to another box | Volatile RAM | Generation of SRTP session keys |
| SRTP Session Key (AES-128) | NIST SP 800-135 KDF | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | For encryption / decryption of SRTP session traffic |
| SRTP Authentication Key (HMAC-SHA1) | Derived from the master key | **Agreement**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | 160-bit HMAC-SHA-1 for message authentication and verification in SRTP |
| SSH Authentication Private Key (RSA) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048 bits) | Virtual Hard Disk | RSA private key for SSH authentication |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | | **Output:** Output as part of HA direct physical connection to another box | | |
| SSH Authentication Public Key (RSA) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048 bits)<br><br>**Output:** Output as part of HA direct physical connection to another box | Virtual Hard Disk | RSA public key for SSH authentication. |
| SSH Session Keys (AES-128, AES-256) | Derived via SSH KDF.<br><br>Note: These keys are generated via SSH (IETF RFC 4251). This protocol enforces limits on the number of total possible encryption/decryption operations. | **Agreement**: Diffie-Hellman | Volatile RAM | Encryption and decryption of SSH session |
| SSH Integrity Keys (HMAC-SHA2-256) | Derived via SSH KDF. | **Agreement**: NA<br><br>**Output:** Output as part of HA direct physical connection to another box | Volatile RAM | 160-bit HMAC-SHA2-256 for message authentication and verification in SSH |
| TLS Authentication Private Key (ECDSA/RSA) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048bits); ECDSA (P- 256/P-384)<br><br>**Output:** Output as part of HA direct physical connection to another box | Virtual Hard Disk | ECDSA/RSA private key for TLS authentication |
| TLS Authentication Public Key (ECDSA/RSA) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048bits); ECDSA (P- 256/P-384)<br><br>**Output:** Output as part of HA direct physical connection to another box | Volatile RAM | ECDSA/RSA public key for TLS authentication. |
| TLS Premaster Secret (48 Bytes) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: NA | Volatile RAM | Establishes TLS master secret |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | | **Entry**: Input during TLS negotiation<br><br>**Output**: Output to peer encrypted by Public Key | | |
| TLS Master Secret (48 Bytes) | Derived from the TLS Pre-Master Secret | **Agreement**: NA | Volatile RAM | Used for computing the Session Key |
| TLS Session Keys (AES-128, AES-256) | Derived from the TLS Master Secret<br><br>Note: These keys are generated via TLS (IETF RFC 5246). This protocol enforces limits on the number of total possible encryption/decryption operations. | **Agreement**: RSA key transport | Volatile RAM | Used for encryption & decryption of TLS session |
| TLS Integrity Keys (HMAC-SHA256 or HMAC-SHA384) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | 160-bit HMAC-SHA256 or HMAC-SHA384 for message authentication and verification in TLS |
| SKEYSEED | Derived by using key derivation function defined in SP800-135 KDF (IKEv2). | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | 160 bit shared secret known only to IKE peers. Used to derive IKE session keys |
| SKEYID (20 Bytes) | Derived by using key derivation function | **Agreement**: NIST SP 800-135 | Volatile RAM | 160 bit secret value used to derive other IKE secrets |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | defined in SP800-135 KDF (IKEv2). | KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | | |
| SKEYID_d<br>(20 Bytes) | Derived using SKEYID, Diffie Hellman shared secret and other non-secret values through key derivation function defined in SP800135 KDF (IKEv1/IKEv2). | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | 160 bit secret value used to derive IKE session keys |
| IKE Pre-Shared Key | Preloaded by the Crypto Officer. | **Agreement**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Flash Memory | Variable size secret used to derive IKE skeyid when using pre-shared secret authentication |
| IKE Session Encryption Key<br>(Triple-DES, AES-128, AES-256 bit) | Derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2) | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | Volatile RAM | Triple-DES, AES 128 or 256 key used to encrypt data |
| IKE Session Authentication Key<br>(HMAC-SHA-512) | Derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2) | **Agreement**: NIST SP 800-135 | Volatile RAM | 512 bit key HMAC-SHA-512 used for data authentication |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | | KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct physical connection to another box | | |
| IKE Private Key (RSA 2048-bit) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048 bits)<br><br>**Output:** Output as part of HA direct physical connection to another box | Volatile RAM | RSA 2048 bit key used to authenticate the module to a peer during IKE |
| IKE Public Key (RSA 2048-bit) | Internal generation by FIPS-approved CTR_DRBG | **Agreement**: RSA (2048 bits)<br><br>**Output:** Output as part of HA direct physical connection to another box | Volatile RAM | RSA 2048 bit public key for TLS authentication. |
| IPsec Session Encryption Key (Triple-DES, AES-128 or AES-256 bit) | Derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct connection to another box | Volatile RAM | Triple-DES, AES 128 or 256 key used to encrypt data |
| IPsec Session Authentication Key (HMAC-SHA-512) | Derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). | **Agreement**: NIST SP 800-135 KDF<br><br>**Entry**: NA<br><br>**Output**: Output as part of HA direct connection to another | Volatile RAM | 512 bit HMAC-SHA-512 key used for data authentication for IPsec traffic |

| CSP Name | Generation/Input | Establishment/ Export | Storage | Use |
|---|---|---|---|---|
| | | box | | |
| Web UI Certificate | Internal generation by FIPS-approved CTR_DRBG | **Agreement:** NA<br><br>**Output:** TLS session with operator | Virtual Hard Disk | Web server certificate |
| Bypass Key (HMAC-SHA-256) | Internal generation by FIPS-approved CTR_DRBG | **Agreement:** NA<br><br>**Output:** NA | Virtual Hard Disk | Bypass service. 256-bit HMAC-SHA-256 used to protect bypass table |

**Table 13: CSP Table**

**Note:** When the module generates symmetric keys or seeds used for generating asymmetric keys, unmodified DRBG output is used as the symmetric key or as the seed for generating the asymmetric keys.

**Note:** All keys generated by the module use the direct output of a FIPS approved DRBG. This meets the requirements of SP 800-133.

# 7. Self-Tests

The modules include an array of self-tests that are run during startup and conditionally during operations to prevent any secure data from being released and to ensure all components are functioning correctly. Self-tests may be run on-demand by power cycling the module.

## 7.1 Power-Up Self-Tests

Acme Packet VME appliance performs the following power-up self-tests when the virtual machine is started. These self-tests require no inputs or actions from the operator:

### 7.1.1 Software integrity Test

- RSA 2048 Software Integrity Test

### 7.1.2 Mocana Cryptographic Library Machine Edition (VME) Self-tests

- AES (Encrypt/Decrypt) Known Answer Test;
- Triple-DES (Encrypt/Decrypt) Known Answer Test;
- SHA-1 Known Answer Test;
- SHA-256 Known Answer Test;
- SHA-384 Known Answer Test;
- SHA-512 Known Answer Test;
- HMAC-SHA-1 Known Answer Test;
- HMAC-SHA-256 Known Answer Test;
- HMAC-SHA-384 Known Answer Test;
- HMAC-SHA-512 Known Answer Test; and
- RSA verify Known Answer Test.

### 7.1.3 Oracle Acme Packet Cryptographic Library Virtual Machine Edition (VME) Self-Tests

- SHA-1 Known Answer Test;
- SHA-256 Known Answer Test;
- SHA-512 Known Answer Test;
- HMAC-SHA-1 Known Answer Test;
- HMAC-SHA-256 Known Answer Test;
- HMAC-SHA-384 Known Answer Test;
- HMAC-SHA-512 Known Answer Test;
- AES (Encrypt/Decrypt) Known Answer Test;
- AES GCM (Encrypt/Decrypt) Known Answer Test;
- SP 800-90A HASH DRBG Known Answer Test;
- SP 800-90A CTR DRBG Known Answer Test;
- RSA sign/verify Known Answer Test; and
- ECDSA sign/verify Known Answer Test.

When the module is in a power-up self-test state or error state, the data output interface is inhibited and remains inhibited until the module can transition into an operational state. While the user may attempt to restart the module to clear an error, the module will require re-installation in the event of a hard error such as a failed self-test.

## 7.2 Critical Functions Self-Tests

Acme Packet VME performs the following critical self-tests. These critical function tests are performed for each SP 800-90A DRBG implemented within the module.

- SP 800-90A Instantiation Test
- SP 800-90A Generate Test
- SP 800-90A Reseed Test
- SP 800-90A Uninstantiate Test

## 7.3 Conditional Self-Tests

The module performs the following conditional self-tests when called by the module:

- Pair Wise consistency tests to verify that the asymmetric keys generated for RSA, and ECDSA work correctly by performing a sign and verify operation;
- Continuous Random Number Generator test to verify that the output of approved-DRBG is not the same as the previously generated value;
- Continuous Random Number Generator test to verify that the output of entropy is not the same as the previously generated value;
- Bypass conditional test using HMAC-SHA-256 to ensure the mechanism governing media traffic is functioning correctly, and;
- Software Load test using a 2048-bit/SHA-256 RSA-Based integrity test to verify software to be updated.

# 8. Crypto-Officer and User Guidance

FIPS Mode is enabled by a license installed by Oracle, which will open/lock down features where appropriate. This section describes the configuration, maintenance, and administration of the cryptographic module.

## 8.1 Secure Setup and Initialization

The operator shall set up the device as defined in the Session Border Controller ACLI Configuration Guide. The Crypto-Officer shall also:

- Verify that the firmware version of the module is Version S-Cz8.2.0 or S-Cz8.2.0p5.
- A new account for the Crypto-Officer and User shall be created as part of Setup and Initialization process. Upon creation of the new CO and User accounts the "default" accounts shipped with the module shall be disabled.
- Ensure all traffic is encapsulated in a TLS, SSH, or SRTP tunnel as appropriate.
- Ensure that SNMP V3 is configured with AES-128/HMAC only.
- Ensure IKEv1 and IKEv2 is using AES CBC or CTR mode for encryption and HMAC-SHA-512 for authentication
- Ensure SSH is configured to use AES CTR mode for encryption.
- Ensure SSH and IKEv1/IKEv2 only use Diffie-Hellman group 14 in FIPS approved mode.
- Ensure all management traffic is encapsulated within a trusted session (i.e., Telnet should not be used in FIPS mode of operation).
- Ensure RSA keys are at least 2048-bit keys for TLS, IKEv1/IKEv2. No 512-bit or 1024-bit keys can be used in FIPS mode of operation.
- All operator passwords must be a minimum of 8 characters in length.
- Ensure use of FIPS-approved algorithms for TLS:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
    - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
    - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
    - TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- Be aware that HA configuration data that contains keys and CSP's must never be transported over an untrusted network. Ensure that the HA ports used for the transport of HA data (including keys and CSP's) are bound to a private IP address range during setup.
- RADIUS and TACACS+ shall not be used in FIPS approved mode.
- HTTPS shall be enabled and configure the web server certificate prior to connecting to the Web UI over TLS.
- Any firmware loaded into this module that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-2 validation.

Services in Table 5 of Section 3.6 make use non-compliant cryptographic algorithms. Use of these algorithms will place the module in a non-Approved mode of operation.

### 8.2 AES-GCM IV Construction/Usage

The AES-GCM IV is used in the following protocols:

- TLS: The TLS AES-GCM IV is generated in compliance with TLSv1.2 GCM cipher suites as specified in RFC 5288 and section 3.3.1 of NIST SP 800-52rev1. Per RFC 5246, when the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key.

In case the module's power is lost and then restored, the key used for the AES GCM encryption or decryption shall be redistributed.

# 9. Mitigation of Other Attacks

The module does not mitigate attacks beyond those identified in FIPS 140-2.

# 10. Operational Environment

## 10.1 Tested Environments

The module is installed using a common base image distributed in a compatible hypervisor format (i.e ova, ovm, qcow2). The software image that is used to deploy the VME is common across all models. The tested configuration includes:

| Operating Environment | Processor | Hardware |
|---|---|---|
| Oracle Linux 7 running on VMware ESXi version 6.5 | Intel Xeon Gold Processor | Oracle Server X7-2 |

**Table 14: Operating environment**

This is considered a modifiable OE as defined by FIPS 140-2. The tested operating environments isolate virtual systems into separate isolated process spaces. Each process space is logically separated from all other processes by the operating environments software and hardware. The module functions entirely within the process space of the isolated system as managed by the single operational environment. This implicitly meets the FIPS 140-2 requirement that only one entity at a time can use the cryptographic module.

## 10.2 Vendor Affirmed Environment

The following platforms have not been tested as part of the FIPS 140-2 level 1 certification however Oracle "vendor affirms" that these platforms are equivalent to the tested and validated platform.  Additionally, Oracle affirms that the module will function the same way and provide the same security services on the system listed below.

| Operating Environment | Processor | Hardware |
|---|---|---|
| Oracle Linux 7 running on VMware ESXi version 6.5 | Intel Xeon Platinum Processors | Oracle Server X7-2 |
| Oracle Linux 7 running on VMware ESXi version 6.5 | Intel Xeon Processor E5-2600 V3 | Oracle Server X5-2 |
| Oracle Linux 7 running on VMware ESXi version 6.5 | Intel Xeon Platinum Processors | Oracle Server X8-2 |

**Table 15: Vendor Affirmed Operating Environment**

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

## Acronyms, Terms and Abbreviations

| Term | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| CDR | Call Data Record |
| CMVP | Cryptographic Module Validation Program |
| CSEC | Communications Security Establishment Canada |
| CSP | Critical Security Parameter |
| DHE | Diffie-Hellman Ephemeral |
| DRBG | Deterministic Random Bit Generator |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| ESBC | Enterprise Session Border Controller |
| EDC | Error Detection Code |
| EMS | Enterprise Management Server |
| HMAC | (Keyed) Hash Message Authentication Code |
| IKE | Internet Key Exchange |
| KAT | Known Answer Test |
| KDF | Key Derivation Function |
| LED | Light Emitting Diode |
| MGT | Management |
| NIST | National Institute of Standards and Technology |
| POST | Power On Self Test |
| PUB | Publication |
| RAM | Random Access Memory |
| ROM | Read Only Memory |
| SHA | Secure Hash Algorithm |
| SNMP | Simple Network Management Protocol |
| SRTP | Secure Real Time Protocol |
| TDM | Time Division Multiplexing |
| TLS | Transport Layer Security |
| VME | Virtual Machine Edition |

**Table 16: Acronyms**

## References

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html.

More information describing the module can be found on the Oracle web site at https://www.oracle.com/industries/communications/enterprise/products/session-border-controller/index.html.

This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is "Oracle - Proprietary" and is releasable only under appropriate non-disclosure agreements.

| Document | Author | Title |
|---|---|---|
| FIPS PUB 140-2 | NIST | FIPS PUB 140-2: Security Requirements for Cryptographic Modules |
| FIPS IG | NIST | Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program |
| FIPS PUB 140-2 Annex A | NIST | FIPS 140-2 Annex A: Approved Security Functions |
| FIPS PUB 140-2 Annex B | NIST | FIPS 140-2 Annex B: Approved Protection Profiles |
| FIPS PUB 140-2 Annex C | NIST | FIPS 140-2 Annex C: Approved Random Number Generators |
| FIPS PUB 140-2 Annex D | NIST | FIPS 140-2 Annex D: Approved Key Establishment Techniques |
| DTR for FIPS PUB 140-2 | NIST | Derived Test Requirements (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules |
| NIST SP 800-67 | NIST | Recommendation for the Triple Data Encryption Algorithm TDEA Block Cypher |
| FIPS PUB 197 | NIST | Advanced Encryption Standard |
| FIPS PUB 198-1 | NIST | The Keyed Hash Message Authentication Code (HMAC) |
| FIPS PUB 186-4 | NIST | Digital Signature Standard (DSS) |
| FIPS PUB 180-4 | NIST | Secure Hash Standard (SHS) |
| NIST SP 800-131A | NIST | Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes |
| PKCS#1 | RSA Laboratories | PKCS#1 v2.1: RSA Cryptographic Standard |

**Table 17: References**