Palo Alto Networks

Prisma SD-WAN Controller's Cryptographic Module

Software Version: 1.0

FIPS 140-2 Level 1 Non-Proprietary Security Policy

Document Version Number: 1.4

Table of Contents

1. Module Overview

Prisma SD-WAN Controller's Cryptographic Module supports the Controller's ability to manage ION devices to administer security policy rules and provide various application and network analytics.

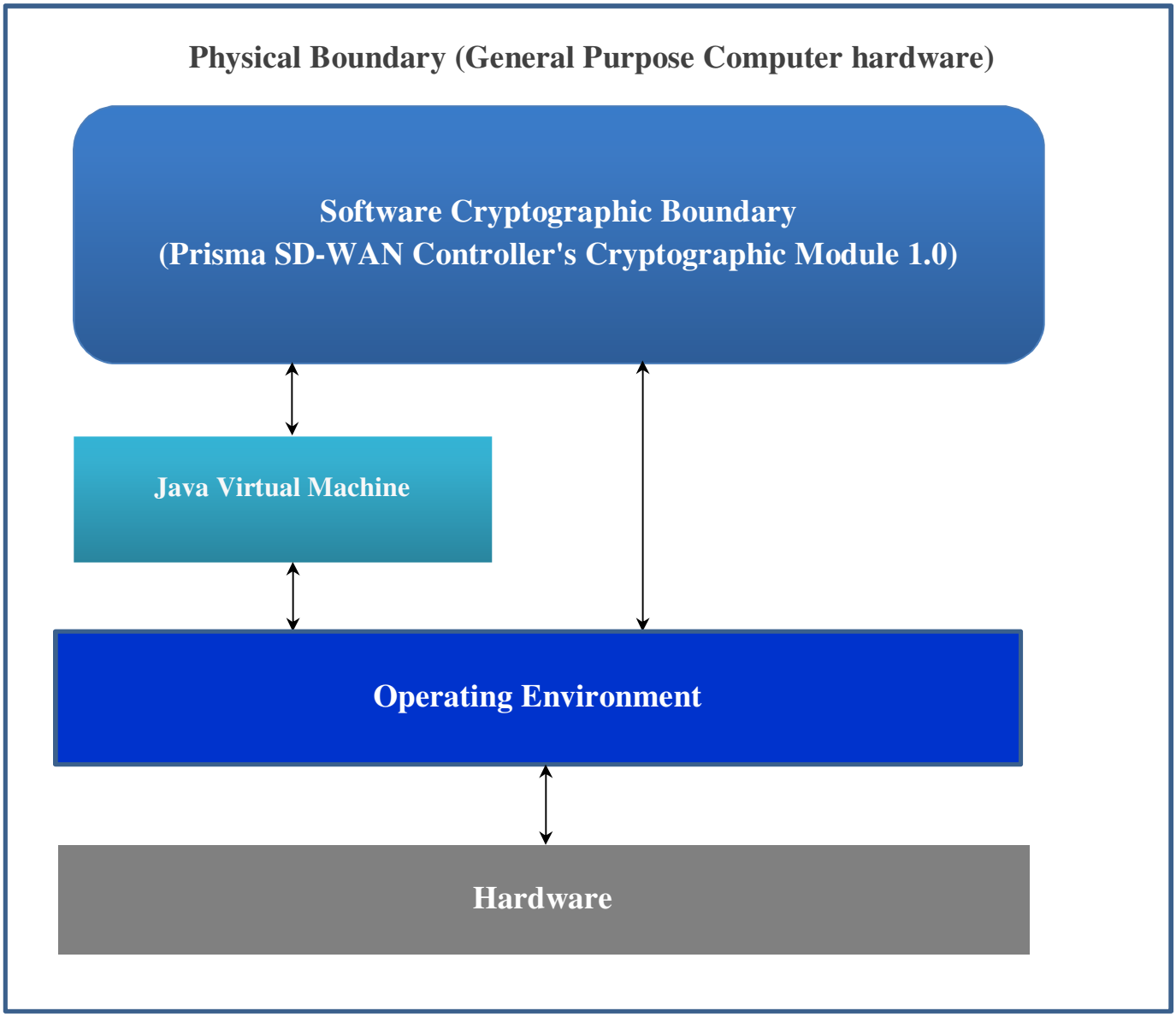
Table 1.1: Configuration tested by the lab

Module	Platform	Processor	Operating Systems
Prisma SD-WAN Controller's Cryptographic Module	Dell Power Edge R740	Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz with and without AES-NI	JDK Version - 11.0.10 on Ubuntu 14.04

Table 1.2: Module Security Level Statement

FIPS Security Area	Security 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

Figure 1: Prisma SD-WAN Controller's Cryptographic Module



2. Modes of Operation

Prisma SD-WAN Controller's Cryptographic Module supports the following two modes of operation to accommodate different operating requirements. The mode is selected implicitly based on the services used.

1) FIPS Approved mode of operation includes functions in Table 2.1.

2) FIPS Non-Approved mode of operation includes functions in Table 2.2.

The installation is performed by authorized personnel with crypto officer role in a secure location which is only accessible by the authorized personnel. The personnel must follow the instructions found in the security policy.

2.1 Approved and Allowed Cryptographic Functions

The following approved cryptographic algorithms are used in FIPS approved mode of operation.

Table 2.1: Approved Cryptographic Functions.

CAVP Cert	Library	Algorithm	Standard	Model/ Method	Key Lengths, Curves or Moduli	Use
A2476	Palo Alto Networks Controller Crypto Library-1	RSA	FIPS 186-4	RSA SigGen PKCS1 v1.5, RSA PSS SHA-224, SHA-256, SHA-384, SHA-512	Mod 2048	Signature Generation
				RSA SigVer PKCS1 v1.5, RSA PSS SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		Signature Verification
		AES	FIPS 197, SP 800-38D	ECB, CBC, GCM[1]	128, 192, 256	Encryption/ Decryption KTS[4]
		HMAC	FIPS 198-1	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	160, 256, 384	TLS Message Authentication Code
		CTR DRBG	SP800-90A	128, 192, 256		Deterministic Random Bit Generation
		ECDSA	FIPS 186-4	ECDSA KeyGen	P-256, P-384, P-521	Key Generation, Key Verification,
				ECDSA KeyVer		

				ECDSA SigGen ECDSA SigVer		Signature Generation, Signature Verification
		SHS	FIPS 180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		TLS Message Digest
		CVL (KDF TLS)	SP800-135	SHA-256, SHA-384, SHA-512		TLS Key Derivation[2]
		KAS-ECC-SSC	SP800-56Ar3	ECC Ephemeral Unified Scheme	P-224,P-256, P-384,P-521 corresponds to 112 to 256 bits of security	TLS Shared Secret Computation
		KAS	SP800-56Ar3 and SP800-135	ECC Ephemeral Unified Scheme	P-224,P-256, P-384,P-521 corresponds to 112 to 256 bits of security	TLS Shared Secret Computation TLS Key Derivation[2]
CKG (vendor Affirmed)			Cryptographic Key Generation			Key Generation[3]
A2496	Palo Alto Networks Controller Crypto Library-2	AES	FIPS 197, SP 800-38D	ECB, CBC, GCM[1]	128, 192, 256	Encryption/ Decryption KTS[4]
		HMAC	FIPS 198-1	HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA-384	160, 256, 384	TLS Message Authentication Code
		SHS	FIPS 180-4	SHA-1, SHA-256, SHA-384		TLS Message Digest
		RSA	FIPS 186-4	RSA KeyGen RSA SigGen PKCS1 v1.5, RSA PSS SHA-224, SHA-256, SHA-384,	Mod 2048	Key Generation, Signature Generation

				SHA-512		
				RSA SigVer PKCS1 v1.5, RSA PSS SHA-224, SHA-256, SHA-384, SHA-512		Signature Verification
		ECDSA	FIPS 186-4	ECDSA KeyGen	P-256, P-384, P-521	Key Generation, Key Verification, Signature Generation, Signature Verification
				ECDSA KeyVer		
				ECDSA SigGen		
				ECDSA SigVer		
		CTR DRBG	SP800-90A	128, 192, 256		Deterministic Random Bit Generation
		Hash DRBG		SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		
		KBKDF	SP800-108	HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512		Key Derivation
		CVL (KDF TLS)	SP800-135	SHA-256, SHA-384, SHA-512		TLS Key Derivation[2]
		KAS-ECC-SSC	SP800-56Ar3	ECC Ephemeral Unified Scheme	P-224,P-256, P-384,P-521 corresponds to 112 to 256 bits of security	TLS Shared Secret Computation
		KAS	SP800-56Ar3 and SP800-135	ECC Ephemeral Unified Scheme	P-224,P-256, P-384,P-521 corresponds to 112 to 256 bits of security	TLS Shared Secret Computation TLS Key Derivation[2]

CKG (vendor Affirmed)			Cryptographic Key Generation			Key Generation[3]

Note 1: Not all CAVS-tested modes of the algorithms are used in this module.

[1]The module's AES-GCM implementation complies with IG A.5 scenario 1 and RFC 5288, and supports acceptable GCM cipher suites from Section 3.3.1 of SP 800-52 Rev 1 or SP 800-52 Rev 2. AES-GCM is only used in TLS version 1.2. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, that encounters this condition will trigger a handshake to establish a new encryption key. New AES-GCM keys are generated by the module if the module loses power.

[2]No parts of this protocol, other than the KDF, has been tested by the CAVP and CMVP.

[3]The module directly uses the output of the DRBG. Section 4, example 1, of SP800-133r2 "Using the Output of a Random Bit Generator" is applicable.

[4] KTS: KTS (AES Certs. #A2476 and #A2496 and HMAC Certs. #A2476 and #A2496; key establishment methodology provides 128 or 256 bits of encryption strength).

Table 2.2: Non FIPS Approved Cryptographic Functions

Algorithm	Use
CHACHA20_POLY1305	Encryption/Decryption
Diffie-Hellman (any modulus)	Key Exchange
sect163k1, sect163r, sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, secp256k1, x25519, x448	Key Exchange Curves
DSA, RSA (less than 2048 bits), ECDSA (curves not equal to P-256, P-384, P-521)	Digital Signatures

3. Ports and interfaces

The physical ports of the module are the same as those of the computer system on which it is executing. The logical interfaces of the module are implemented via an Application Programming Interface (API). The following table describes each logical interface.

Table 3: FIPS 140-2 Logical Interfaces

Logical Interface	Description
Data Input	Input parameters that are supplied to the API commands
Data Output	Output parameters that are returned by the API commands

Logical Interface	Description
Control Input	API commands
Status Output	Return status provided by API commands

4. Roles and Services

The module supports the following roles:

User role: The user uses the cryptographic services provided by the module.

Crypto Officer role: The Crypto Officer installs and manages the module.

Table 4: Roles and Services

Service	Corresponding Roles	Types of Access to Cryptographic Keys and CSPs R – Read or Execute W – Write or Create Z – Zeroize
Installation	Crypto Officer	N/A
Initialize	Crypto Officer	N/A
Self-test	Crypto Officer	N/A
Show status	Crypto Officer User	N/A
Zeroization	Crypto Officer	All:Z
Reboot or shutdown	Crypto Officer	N/A
Client TLS connect	Crypto Officer User	TLS Keys: R,W DRBG seed: R, W
Certificate Generation	Crypto Officer User	DRBG seed: R, W CA Certificate and Private Key: R, W User Certificate: R,W
VPN Key Generation	Crypto Officer User	DRBG seed: R, W VPN Key: R,W
Device TLS connect	Crypto Officer User	TLS Keys: R,W DRBG seed: R, W
External CA TLS connect	Crypto Officer User	TLS Keys: R,W DRBG seed: R, W

Non-Approved services are implementations of non FIPS Approved Cryptographic Functions. They are listed in Table 2.2.

5. Cryptographic Keys and CSPs

The table below describes the cryptographic keys and CSPs used by the module.

Table 5: Cryptographic Keys and CSPs

Key	Description/Usage	Storage
TLS master secret Established using KDF TLS	Used to derive TLS encryption key and TLS HMAC Key	RAM in plaintext
TLS pre-master secret Established using KAS-ECC-SSC	Used to derive TLS master Secret	RAM in plaintext
TLS AES key Established using KDF TLS	Used during encryption and decryption of data within the TLS protocol	RAM in plaintext
TLS HMAC key Established using KDF TLS	Used to protect integrity of data within the TLS protocol	RAM in plaintext
TLS RSA public and private keys Established using DRBG or set by operators	Used during the TLS handshake	RAM in plaintext
TLS ECDSA public and private keys Established using DRBG or set by operators	Used during the TLS handshake	RAM in plaintext

Key	Description/Usage	Storage
TLS ECC Diffie-Hellman SP800-56Ar3 public and private keys Established using DRBG	Used during the TLS handshake to establish the shared secret	RAM in plaintext
CTR_DRBG CSPs: entropy input, V and Key Hash DRBG: entropy input, V and C Entropy is loaded externally	Used during generation of random numbers	RAM in plaintext
CA Certificate and Private Key Established using DRBG	Used during end user enrollment	RAM in plaintext
User Certificate Established using DRBG	Used during end user enrollment	RAM in plaintext
VPN Key Established using KBKDF	Used during VPN enrollment	RAM in plaintext

Note 1: public keys are not considered CSPs

Note 2: All keys that are generated by this module are generated by using the DRBG. Entropy is loaded externally. Minimum number of bits of entropy loaded is 256-bits, since the minimum length of the entropy field is at least 256-bits.

Note 3: Keys can be provided to the module via API input parameters and output via API output parameters. The module does not enter or output keys outside its physical boundary. Zeroization is performed using power cycle.

6. Self-tests

The module performs the following power-up self-tests when the module is started or restarted. Upon failure or a power-up self-test the module halts its operation.

Table 6: Self-Tests

Algorithm	Test
Software integrity	HMAC-SHA2-256
AES	KAT (CBC / GCM encryption/decryption are separately tested)
KAS (ECC-SSC)	KAT per implementation guidance
	ECC DH Private/Public Key Validation tests as per SP800-56Ar3
ECDSA	KAT (curve sizes P-224 and K-233) using SHA-256
	Pairwise Consistency Test
HMAC	KAT (HMAC-SHA-1/224/256/384/512)
KBKDF	KAT
DRBG	KAT
	Continuous Random Number Generator test
	DRBG health tests
TLS 1.2 KDF	KAT
RSA	KAT (key size tested: 2048, using SHA-256)
	Pairwise Consistency Test
SHA	KAT (SHA-1/224/256/384/512)

7. References

Table 7: References

Reference	Specification
[ANS X9.31]	Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)
[FIPS 140-2]	Security Requirements for Cryptographic modules, May 25, 2001
[FIPS 180-4]	Secure Hash Standard (SHS)
[FIPS 186-2/4]	Digital Signature Standard
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[FIPS 202]	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

Reference	Specification
[PKCS#1 v2.1]	RSA Cryptography Standard
[PKCS#5]	Password-Based Cryptography Standard
[PKCS#12]	Personal Information Exchange Syntax Standard
[SP 800-38A]	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode
[SP 800-38B]	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication
[SP 800-38C]	Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality
[SP 800-38D]	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC
[SP 800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
[SP 800-56A]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[SP 800-56B]	Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography
[SP 800-56C]	Recommendation for Key Derivation through Extraction-then-Expansion
[SP 800-67R1]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-89]	Recommendation for Obtaining Assurances for Digital Signature Applications
[SP 800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[SP 800-108]	Recommendation for Key Derivation Using Pseudorandom Functions
[SP 800-132]	Recommendation for Password-Based Key Derivation
[SP 800-135]	Recommendation for Existing Application –Specific Key Derivation Functions