Hewlett Packard Enterprise

Technical Brief

HPE Aruba Networking EdgeConnect Enterprise Cryptographic Algorithms

ECOS 9.2 and Orchestrator 9.2

Important notice

Revision A, August 2022

Open Source Code:

Hewlett Packard Enterprise Company
Attn: General Counsel
WW Corporate Headquarters
1701 E Mossy Oaks Rd Spring, TX 77389
United States of America.

HPE aruba networking

Contents

Introduction

This document covers all cryptographic suites used by Aruba EdgeConnect Enterprise appliances including management plane connections to Orchestrator and Cloud Portal, and data plane connections between EdgeConnect Enterprise appliances as well as third-party devices. Figure 1 depicts all management plane and data plane connections to EdgeConnect Enterprise appliances.

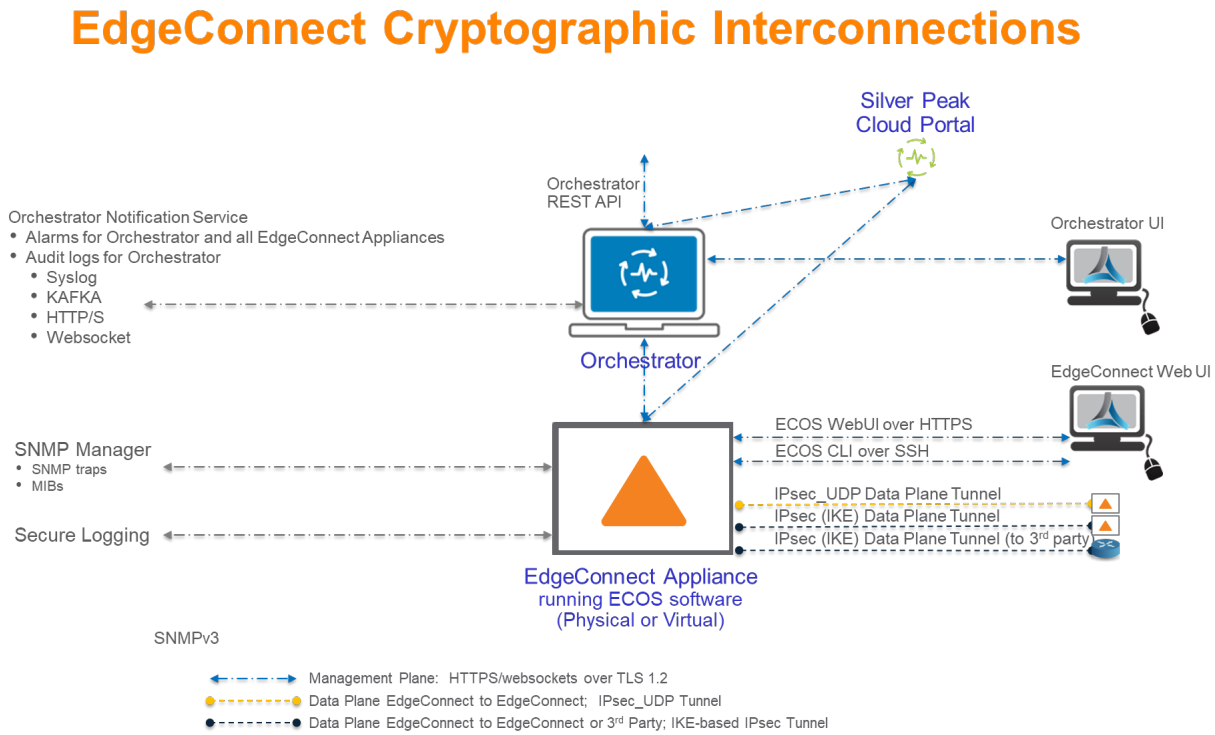


Figure 1 EdgeConnect Enterprise Cryptographic Interconnections

Management Plane Interfaces

The following management interfaces can be used to configure or provision EdgeConnect Enterprise appliances running ECOS software.

Orchestrator User Interface

Orchestrator acts as an aggregation layer for configuration, provisioning, monitoring, and reporting for all EdgeConnect Enterprise appliances in an SD-WAN. Administrative users interact with the Orchestrator User Interface to orchestrate the SD-WAN, manage templates across all or subsets of appliances, configure site-specific parameters on EdgeConnect Enterprise appliances.

Orchestrator API

For machine-to-machine systems integration, Orchestrator provides a RESTful Application Programming Interface (API). Using the API, configuration actions (POST, PUT, DELETE) and queries (GET) can be applied to all appliances or a single appliance using the NE Primary Key (NE.Pk) as an identifier. The REST API runs over a secure WebSocket connection using TLS 1.2 between the Orchestrator and the ECOS appliance.

Orchestrator Notification Service

Orchestrator Notification Service provides real-time streaming of EdgeConnect Enterprise and Aruba Orchestrator alarms or Orchestrator logs. Depending on the protocol, Orchestrator will act as server or client.

ECOS Web User Interface

Administrative users can use the Web User Interface (Web UI), which provides a highly intuitive, graphical interface for the comprehensive set of ECOS capabilities. The Web UI can be accessed from a TLS-enabled web browser using HTTPS on logical port 443. It is always recommended to use Orchestrator for all provisioning and configuration.

NOTE: It is strongly recommended that HTTP be disabled.

ECOS CLI Over SSH

Administrative users can use the CLI to perform security-sensitive and non-security-sensitive monitoring and configuration. The CLI can be accessed remotely by using the SSHv2 secured management session over the Ethernet ports. CLI is typically only used for diagnostic and/or troubleshooting. CLI should not be used for general provisioning, all provisioning/configuration should be performed on Orchestrator.

ECOS Appliance-Level API

EdgeConnect Enterprise appliances running ECOS software provide an appliance-level API. This interface is used for all configuration, provisioning, monitoring, and reporting between the Orchestrator and the ECOS appliance. Administrative users typically do not access the ECOS API.

ECOS Secure Logging

Starting with ECOS 9.2, EdgeConnect Enterprise appliances provide secure transport to remote log receivers, providing the flexibility of shipping logs using TCP_SSL (TLSv1.2). It provides configurable options:

1. Upload client certificate for client to authenticate itself with the remote log receiver.
2. Verify the server certificate.
3. Flexibility of configuring any port.

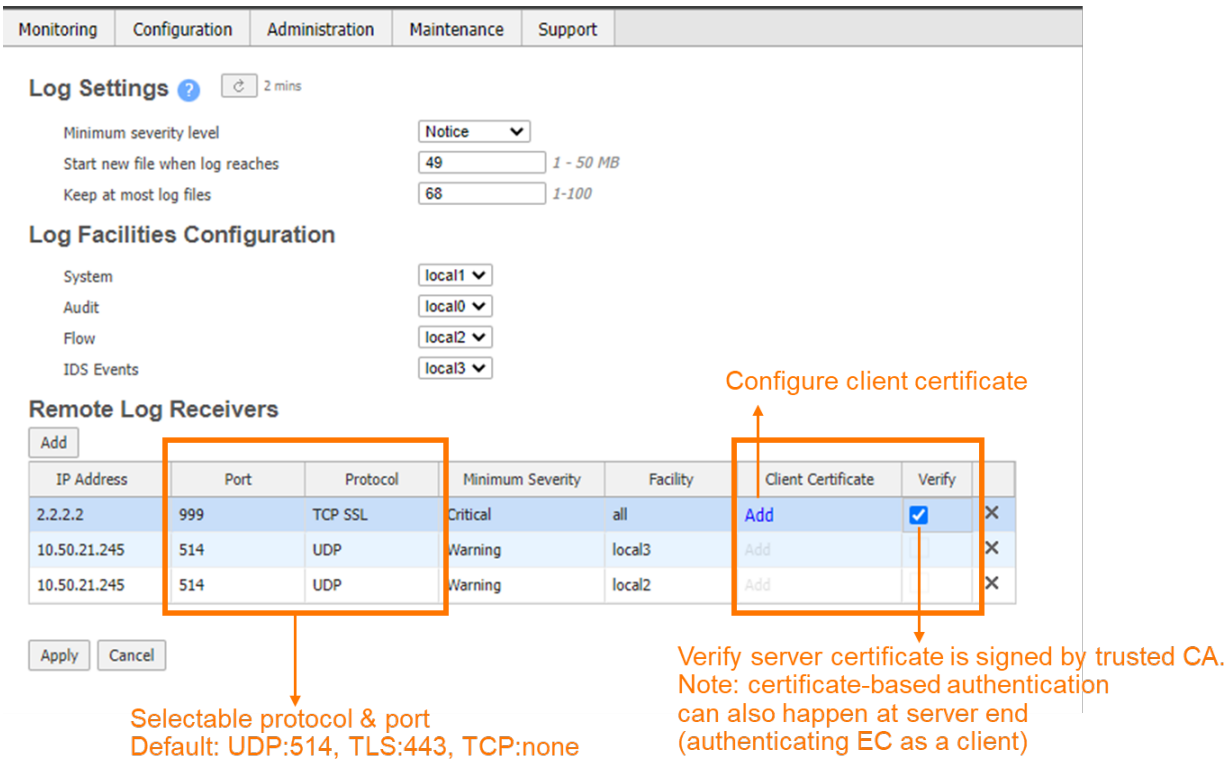


Figure 2 ECOS Secure Logging

Peer certificate authentication options are summarized here.

Table 1: Summary of ECOS Secure Logging Options

Client Certificate Uploaded	Verify Option Selected	Behavior
No	No	Works fine if server is not configured to request client certificate. EC does not verify if server certificate is from trusted CA.
Yes	No	EC shares client cert with server during TLS handshake.
No	Yes	EC will verify server certificate is from trusted CA. If custom CA is used by server, but custom CA is not configured, connection will fail. Additionally, if server is configured to verify client, but no client certificate is configured, connection will fail.
Yes	Yes	EC shares client cert with server during TLS handshake. EC will verify server certificate is from trusted CA. If custom CA is used by server, but custom CA is not configured, connection will fail.

Cipher Suites

There are 4 parts in the cipher name.

1. Key exchange algorithm: for example, ECDHE
2. Authentication algorithm: for example, RSA
3. Symmetric encryption algorithm: AES-CBC mode 128 bits or AES-GCM mode 128 bits
4. Message authentication code (MAC) algorithm: SHA-256

Examples of common ciphers used in ECOS for TLS 1.2:

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Cipher input string syntax follows the format specified by openssl.org at:

https://www.openssl.org/docs/man1.0.2/man1/ciphers.html

EdgeConnect Enterprise Cryptographic Algorithms

ECOS TLS as Server (ECOS Web UI)

Starting with ECOS 9.0.3 and Orchestrator 9.0.4, the Session Management template also provides a way to directly enter an OpenSSL Cipher List, if required by the enterprise customer security team. This is intended to be used only by enterprise customers who have strict security requirements and want to set their own ciphers for SSH and Web UI (TLS server). Orchestrator provides the Session Management template option that orchestrates the cipher setting across all appliances with one click. Figure 3 shows the template user interface with the help window expanded. A warning informs customers to test the proposed cipher in their environment to ensure compatibility with connected systems.

NOTE: Aruba recommends leaving the OpenSSL Cipher List field blank (default).

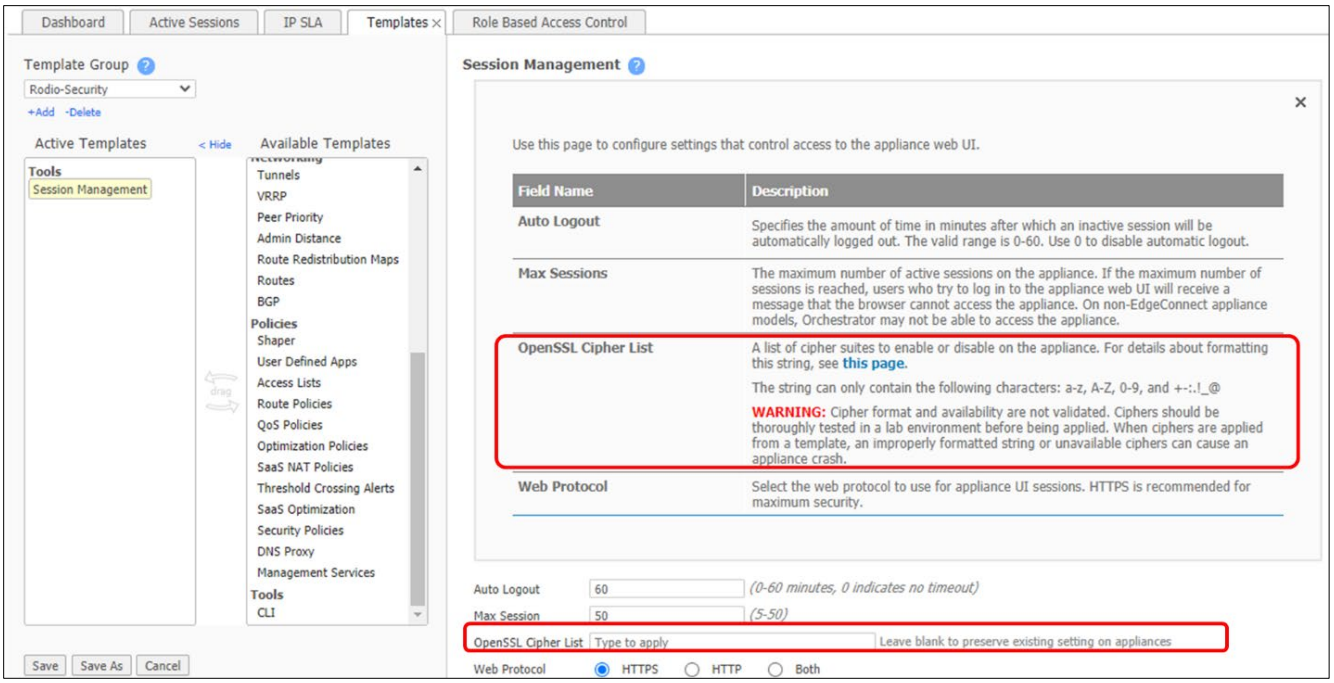


Figure 3: Session Management Template Help Window

Cipher Suite supported on ECOS WebUI:

Table 2 ECOS WebUI Cipher Suite

“Out of the box” Cipher Suites	ECOS	Curve
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	All releases	(secp256r1)
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	All releases	(secp256r1)
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	All releases	(secp256r1)
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	All releases	(secp256r1)
TLS_RSA_WITH_AES_128_CBC_SHA	8.1.9 (see Note 1)	(RSA 2048)
TLS_RSA_WITH_AES_128_CBC_SHA256	8.1.9	(RSA 2048)

"Out of the box" Cipher Suites	ECOS	Curve
TLS_RSA_WITH_AES_128_GCM_SHA256	8.1.9	(RSA 2048)
TLS_RSA_WITH_AES_256_CBC_SHA	8.1.9	(RSA 2048)
TLS_RSA_WITH_AES_256_CBC_SHA256	8.1.9	(RSA 2048)
TLS_RSA_WITH_AES_256_GCM_SHA384	8.1.9	(RSA 2048)

Note 1: ECOS 8.3.2, 9.0, and 9.1 remove the support of RSA as key exchange mechanism (removal of Static Key Ciphers). Starting with ECOS 9.0.2, ECOS WebUI cipher list is provisionable Orchestrator templates.

Data Plane (IPsec)

EdgeConnect to EdgeConnect Underlay Tunnels, IPsec UDP

Peer Authentication	ECOS	Note
Proprietary	All releases	See below

For EdgeConnect Enterprise appliances authenticate with their associated Orchestrator, newly installed EdgeConnect Enterprise devices must first establish trust with Silver Peak's Cloud Portal. Once trust is established with Cloud Portal, trust is established between the EdgeConnect Enterprise appliance and its Orchestrator.

Aruba hardware appliances authenticate to Cloud Portal with a built-in shared secret. Aruba virtual appliances require the account name and account key.

Key Exchange	ECOS	Note
Proprietary	All releases	See below

EdgeConnect Enterprise appliances derive encryption keys for IPsec UDP tunnels by cryptographically combining two components:

- Orchestrator-derived ephemeral key material
- EdgeConnect nonce

The derived encryption keys used in IPsec UDP tunnels are never sent on the wire (data plane or management plane). They are independently and uniquely derived per EdgeConnect Enterprise appliance, UT, and direction. Orchestrator generates ephemeral key material, which is a time-varying "seed" that distributes the same key material to all appliances. Therefore, the key material is global to all appliances in the SD-WAN but varies by time.

NOTE: Orchestrator provides key material and not the encryption key itself. Key material lifetime, rotation period, and appliance storage/persistence are managed by Orchestrator.

EdgeConnect Enterprise appliances create and persist a nonce per UT, per direction. Therefore, the nonce is static over the lifetime of the tunnel but varies by UT and direction. When combined, every UT and direction has globally unique, time-varying encryption keys.

Encryption (See Note 2)	ECOS	Mode
"Auto" is default on Orchestrator UI. **"Auto" for IPsec UDP = AES_128_CBC**	All releases	CBC
AES_128_CBC	All releases	CBC

Encryption (See Note 2)	ECOS	Mode
AES_256_CBC	All releases	CBC
AES_128_GCM_16	ECOS 9.2	GCM
AES_256_GCM_16	ECOS 9.2	GCM

Note 2: GCM mode includes authentication.

Message Authentication	ECOS	Integrity Check Value (ICV) (Authentication Tag) Length
SHA-1 (Default)	All releases	12 octets
SHA-2 256	All releases	16 octets
SHA-2 384	All releases	24 octets
SHA-2 512	All releases	32 octets
AES_128_GCM	ECOS 9.2	16 octets
AES_256_GCM	ECOS 9.2	16 octets
AES_128_GMAC	ECOS 9.2	16 octets
AES_256_GMAC	ECOS 9.2	16 octets

EdgeConnect to EdgeConnect Underlay and Passthrough Tunnels, IKE-based IPsec

Key Exchange	ECOS	Peer Authentication
IKEv1 **(Default in all releases < ECOS 9.2.1)**	Available in All releases	Pre-shared Key
IKEv2	Available starting with ECOS 8.3+	Pre-shared Key
IKEv2	Default starting with ECOS 9.2.1	Pre-shared Key

IKE Encryption (IKEv1)	AutoOrder selection	ECOS
"Auto" is default on Orchestrator UI and appliance webUI.	Encryption algorithm is negotiated by both ends in the order prescribed below (1 – first; 4 – last)	All releases
AES_128_CBC	2	All releases
AES_256_CBC	1	All releases
IKE Encryption (IKEv1)	AutoOrder selection	ECOS

For IKEv1-based IPsec, Auto means AES_128_CBC or AES_256_CBC are negotiated; the maximum common key lengths available on both peers is jointly selected in preference order listed above.

IKE Encryption (IKEv2)	AutoOrder selection	ECOS
"Auto" is default on Orchestrator UI and appliance webUI.	Encryption algorithm is negotiated by both ends in the order prescribed below (1 – first; 4 – last)	All releases
AES_128_CBC	4	All releases
AES_256_CBC	3	All releases
AES_128_GCM_16	2	ECOS 9.2.0+
AES_256_GCM_16	1	ECOS 9.2.0+

For IKEv2-based IPsec, Auto means AES_128_CBC, AES_256_CBC, AES_128_GCM, AES_256_GCM are negotiated; the maximum common key lengths available on both peers is jointly selected in preference order listed above.

IKE Authentication (See Note 3)	IKE Version	ECOS	Integrity Check Value (ICV) (Authentication Tag) Length
SHA_1 (Default)	v1 or v2	All releases	12 octets
SHA_2_256	v1 or v2	All releases	16 octets
SHA_2_384	v1 or v2	All releases	24 octets
SHA_2_512	v1 or v2	All releases	32 octets

Note 3: When AES_128_GCM or AES_256_GCM are selected for encryption, these algorithms support Authentication.

IKE (phase 1) Diffie-Hellman Groups	Tunnel Type	ECOS	Length (bits)
Regular Modular Prime Groups			Modulus
Diffie-Hellman Group 1	PT only	All releases	768
Diffie-Hellman Group 2 (default for PT)	PT only	All releases	1024
Diffie-Hellman Group 5	PT only	All releases	1536
Diffie-Hellman Group 14 (default for UT)	UT and PT	All releases	2048
Diffie-Hellman Group 15	UT and PT	All releases	3072
Diffie-Hellman Group 16	UT and PT	All releases	4096
Diffie-Hellman Group 17	UT and PT	All releases	6144
Diffie-Hellman Group 18	UT and PT	All releases	8192
NIST Elliptic Curve Groups			Prime Size
Diffie-Hellman Group 19	UT and PT	ECOS 9.2.0+	256
Diffie-Hellman Group 20	UT and PT	ECOS 9.2.0+	384
Diffie-Hellman Group 21	UT and PT	ECOS 9.2.0+	521
Diffie-Hellman Group 26	UT and PT	ECOS 9.2.0+	224
Modern Elliptical Curve Groups			Modulus
Diffie-Hellman Group 31	UT and PT	ECOS 9.2.0+	256

IKE Pseudo-Random Functions	When Selectable	ECOS
SHA_2_256	AES_GCM_128 only	All releases
SHA_2_384		All releases
SHA_2_512		All releases

IPsec Encryption	ECOS	Mode	Authentication Tag
“Auto” is default on Orchestrator UI and appliance webUI. **Auto means AES_128_CBC or AES_256_CBC is negotiated; the maximum common key lengths available on both peers is jointly selected.**	All releases	CBC	16 octets

IPsec Encryption	ECOS	Mode	Authentication Tag
AES_128_CBC	All releases	CBC	16 octets
AES_256_CBC	All releases	CBC	16 octets
AES_128_GCM_16	ECOS 9.2	GCM	16 octets
AES_256_GCM_16	ECOS 9.2	GCM	16 octets

IPsec Authentication	ECOS	Message Length
SHA_1 (Default)	All releases	
SHA_2_256	All releases	
SHA_2_384	All releases	
SHA_2_512	All releases	
AES_128_GMAC	ECOS 9.2	
AES_256_GMAC	ECOS 9.2	

IPsec (phase 2) Perfect Forwarding Secrecy Groups	Tunnel Type	ECOS	Length (bits)
None (Default), see Note 4	UT and PT	All releases	
Regular Modular Prime Groups			Modulus
Diffie-Hellman Group 1	PT only	All releases	768
Diffie-Hellman Group 2	PT only	All releases	1024
Diffie-Hellman Group 5	PT only	All releases	1536
Diffie-Hellman Group 14	UT and PT	All releases	2048
Diffie-Hellman Group 15	UT and PT	All releases	3072
Diffie-Hellman Group 16	UT and PT	All releases	4096
Diffie-Hellman Group 17	UT and PT	All releases	6144
Diffie-Hellman Group 18	UT and PT	All releases	8192
NIST Elliptic Curve Groups			Prime Size
Diffie-Hellman Group 19	UT and PT	ECOS 9.2.0+	256
Diffie-Hellman Group 20	UT and PT	ECOS 9.2.0+	384
Diffie-Hellman Group 21	UT and PT	ECOS 9.2.0+	521
Diffie-Hellman Group 26	UT and PT	ECOS 9.2.0+	224
Modern Elliptical Curve Groups			Modulus
Diffie-Hellman Group 31	UT and PT	ECOS 9.2.0+	256

Note 4: if set to none, no negotiation recurs during phase2 and the same DH group is used from the IKE phase.

Boost Network Memory Disk Encryption

ECOS All Releases	Mode
AES_128	CBC

ECOS software upgrade

ECOS upgrade packages require SHA 256 image verification.

SNMPv3

Encryption	ECOS	Mode
AES_128 (Default)	All releases	CBC

Hash	ECOS
SHA-1 (Default)	All releases

SSHv2

Starting with ECOS 9.0.2, SSH cipher list is provisionable on ECOS CLI.

Encryption	ECOS	Mode
AES_128 (Default)	All releases	CTR
AES_192	All releases	CTR
AES_256	All releases	CTR

Hash	ECOS
SHA-1 (Default)	All releases

Key Exchange	ECOS	length
Diffie-Hellman Group 14 (Default)	All releases	2048 bit

ECOS Secure Logging

ECOS R9.2 introduces the option for secure logging.

SSL Acceleration

By supporting the use of SSL certificates and keys, EdgeConnect Enterprise's Boost feature provides deduplication for Secure Socket Layer (SSL) encrypted WAN traffic:

- ECOS decrypts SSL data using the configured certificates and keys, optimizes the data, and transmits data over an IPSec tunnel. The peer EdgeConnect Enterprise appliance uses configured SSL certificates to re-encrypt data before transmitting towards the LAN.
- Peers that exchange and optimize SSL traffic must use the same certificate and key.
- If the enterprise certificate that is used for signing substitute certificates is subordinate to higher level Certificate Authorities (CA), then those CA certificates must be added. If the browser cannot validate up the chain to the root CA, it will issue a warning that it cannot trust the certificate.
- Orchestrator templates are used to provision a certificate and its associated key across multiple appliances.
    - Both PFX certificates (generally, for Microsoft servers) or PEM certificates are supported.
    - The default is PEM when PFX Certificate File is deselected.

- If the key file has an encrypted key, then the passphrase needed to decrypt it must be provisioned.

SSL Acceleration Protocol Support	ECOS
SSL v3	All releases
TLS 1.0	All releases
TLS 1.1	All releases
TLS 1.2	All releases

Key Exchange	ECOS
RSA	All releases
DHE_RSA	All releases
ECDHE_RSA	All releases

Authentication	ECOS
RSA	All releases
ECDSA	All releases

Encryption	ECOS	Mode
AES_128	All releases	CTR
AES_256	All releases	CTR
AES_128	All releases	GCM
AES_256	All releases	GCM
RC4	All releases	
3DES	All releases	

Message Authentication	ECOS
MD5	All releases
SHA-1	All releases
SHA-2 384	All releases

Orchestrator Cryptographic Algorithms

TLS Cipher Suite Supported on Orchestrator

Orchestrator acts as server for the cases listed here:

- User Interface (browser is client)
- EdgeConnect Enterprise Appliance management plane (ECOS is client)
- Rest API
- Remote Log Receiver – Websocket

Orchestrator as server choses the strongest crypto cipher suite from the intersection between server and client.

Orchestrator acts as client for the cases listed here:

- Connection to Cloud Portal
- 3rd party REST API services (e.g., Zscaler, AWS, Azure)
- Google Maps to convert address to Lat/Lon
- Remote Log Receiver – HTTPS

Whether Orchestrator is client or server, it supports the Cipher Suites listed in the table below.

Table 3: Orchestrator UI Cipher Suites Supported for all Releases

Key Exchange	Authentication	Symmetric Encryption	Key Length	AES Mode	Message Authentication	Combined Cipher String
DHE	DSS	AES	128	CBC	SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
DHE	DSS	AES	128	CBC	SHA256	TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
DHE	DSS	AES	128	GCM	SHA256	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
DHE	DSS	AES	256	CBC	SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
DHE	DSS	AES	256	CBC	SHA256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
DHE	DSS	AES	256	GCM	SHA384	TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
DHE	RSA	AES	128	CBC	SHA	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
DHE	RSA	AES	128	CBC	SHA256	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
DHE	RSA	AES	128	GCM	SHA256	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
DHE	RSA	AES	256	CBC	SHA	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
DHE	RSA	AES	256	CBC	SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
DHE	RSA	AES	256	GCM	SHA384	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

Key Exchange	Authentication	Symmetric Encryption	Key Length	AES Mode	Message Authentication	Combined Cipher String
ECDH	ECDSA	AES	128	CBC	SHA	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
ECDH	ECDSA	AES	128	CBC	SHA256	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
ECDH	ECDSA	AES	128	GCM	SHA256	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
ECDH	ECDSA	AES	256	CBC	SHA	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
ECDH	ECDSA	AES	256	CBC	SHA384	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
ECDH	ECDSA	AES	256	GCM	SHA384	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
ECDH	RSA	AES	128	CBC	SHA	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
ECDH	RSA	AES	128	CBC	SHA256	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
ECDH	RSA	AES	128	GCM	SHA256	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
ECDH	RSA	AES	256	CBC	SHA	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
ECDH	RSA	AES	256	CBC	SHA384	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
ECDH	RSA	AES	256	GCM	SHA384	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
ECDHE	ECDSA	AES	128	CBC	SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
ECDHE	ECDSA	AES	128	CBC	SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
ECDHE	ECDSA	AES	128	GCM	SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE	ECDSA	AES	256	CBC	SHA	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ECDHE	ECDSA	AES	256	CBC	SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE	ECDSA	AES	256	GCM	SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE	RSA	AES	128	CBC	SHA	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
ECDHE	RSA	AES	128	CBC	SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE	RSA	AES	128	GCM	SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE	RSA	AES	256	CBC	SHA	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ECDHE	RSA	AES	256	CBC	SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE	RSA	AES	256	GCM	SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

Key Exchange	Authentication	Symmetric Encryption	Key Length	AES Mode	Message Authentication	Combined Cipher String
See Note 5	RSA	AES	256	CBC	SHA	TLS_RSA_WITH_AES_256_CBC_SHA
	RSA	AES	128	CBC	SHA	TLS_RSA_WITH_AES_128_CBC_SHA
	RSA	AES	256	CBC	SHA256	TLS_RSA_WITH_AES_256_CBC_SHA256
	RSA	AES	128	CBC	SHA256	TLS_RSA_WITH_AES_128_CBC_SHA256
	RSA	AES	128	GCM	SHA256	TLS_RSA_WITH_AES_128_GCM_SHA256
	RSA	AES	256	GCM	SHA384	TLS_RSA_WITH_AES_256_GCM_SHA384

Note 5: RSA performs both Key Exchange and Authentication.

Orchestrator SSH Cipher Suite

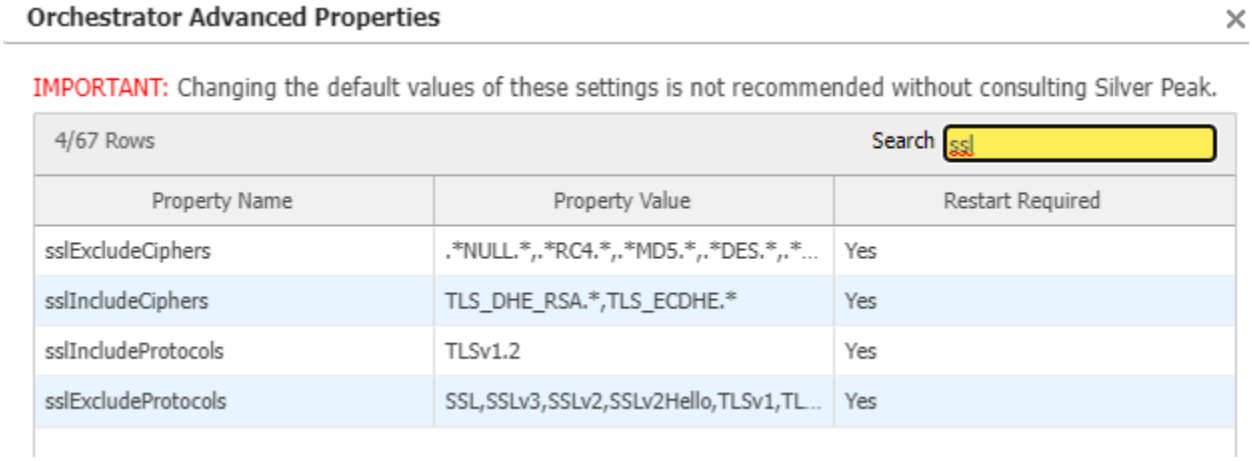
Algorithm	Cipher Suite
kex_algorithms	
	curve25519-sha256
	ecdh-sha2-nistp256
	ecdh-sha2-nistp384
	ecdh-sha2-nistp521
	diffie-hellman-group-exchange-sha256
	diffie-hellman-group16-sha512
	diffie-hellman-group18-sha512
	diffie-hellman-group-exchange-sha1
	diffie-hellman-group14-sha256
	diffie-hellman-group14-sha1
	diffie-hellman-group1-sha1
server_host_key_algorithms	
	ssh-rsa

Algorithm	Cipher Suite
	rsa-sha2-256
	rsa-sha2-512
	ecdsa-sha2-nistp256
	ssh-ed25519
encryption_algorithms	
	aes256-ctr
	aes192-ctr
	aes128-ctr
	aes256-cbc
	aes192-cbc
	aes128-cbc
mac_algorithms	
	hmac-sha1
	hmac-sha2-256
	hmac-sha2-512

Orchestrator Advanced Properties

Orchestrator allows configuration of Allowed and Disabled Cipher Suites.

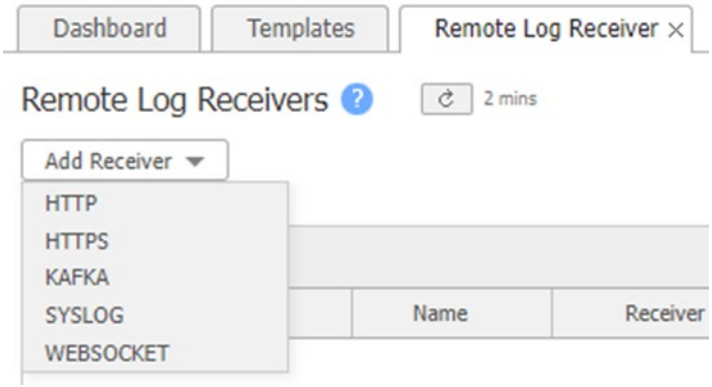
- Self-hosted Orchestrators: end-customers can modify these directly.
- Aruba-hosted, Orchestrator-as-a-Service: Technical Assistance Center (TAC) request is required.



NOTE: Aruba recommends leaving these cipher suites as default. Any changes must be thoroughly tested for compatibility.

Orchestrator Remote Log Receiver Connections

Orchestrator Notification Service (ONS) provides the facility to stream Orchestrator and EdgeConnect Enterprise Alarms and Orchestrator log events to third-party systems. Protocol options include HTTP, HTTPS, KAFKA, SYSLOG, and WEBSOCKET.



EdgeConnect Enterprise (ECOS) Cryptographic Key Management and Critical Security Parameters

Item	Description/Usage	Description/Usage	Type
TLS as Server (ECOS Web UI)			
1	TLS pre-master secret	Used to derive TLS master secret	Length: 48 bytes
2	TLS master secret	Used to derive TLS encryption key and TLS HMAC Key	Length: 48 bytes
3	TLS AES key	Used during encryption and decryption of data within the TLS protocol	AES-CBC (128, 256 bits) AES-GCM (128, 256 bits) (192 bits not supported)
4	TLS HMAC key	Used to protect integrity of data within the TLS protocol	Using SHA1, SHA256, SHA384
5	TLS server RSA public key	Used during the TLS handshake	2048 bits or larger
6	TLS server RSA private key	Used during the TLS handshake	2048 bits or larger
7	TLS client RSA public key	TLS client sends the public key	2048 bits, 3072 bits
8	TLS Server EC Diffie-Hellman public key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
9	TLS Server EC Diffie-Hellman private key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
10	TLS Client EC Diffie-Hellman public key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
11	TLS KDF internal states	Used to derive master secret from pre-master secret. Use to derive key materials from master secret.	NIST SP800-135 TLS KDF
TLS as Client (ECOS Client of Orchestrator, Cloud Portal, and image server)			
12	TLS pre-master secret	Used to derive TLS master secret	Length: 48 bytes
13	TLS master secret	Used to derive TLS encryption key and TLS HMAC Key	Length: 48 bytes

Item	Description/Usage	Description/Usage	Type
14	TLS AES key	Used during encryption and decryption of data within the TLS protocol	AES-CBC (128, 256 bits) AES-GCM (128, 256 bits) (192 bits not supported)
15	TLS HMAC key	Used to protect integrity of data within the TLS protocol	Using SHA1, SHA256, SHA384
16	TLS Server RSA public key	Used during the TLS handshake	2048 bits or larger
17	TLS Client EC Diffie-Hellman public key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
18	TLS Client EC Diffie-Hellman private key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
19	TLS Server EC Diffie-Hellman public key	Used during the TLS handshake to establish the shared secret	EC DH P-224, P-256, P-384, P-521 B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571
20	TLS KDF internal states	Used to derive master secret from pre-master secret. Use to derive key materials from master secret.	SP800-135 TLS KDF
21	Trusted root CA public keys	Used to authenticate the TLS server	RSA 2048 bits, 4096 bits EC DH P-256, P-384 ECDSA SHA1, SHA256, SHA384, SHA512
Passwords			
22	CO and User Login Passwords	Used for operator authentication to the CLI, WebUI	8 to 64 character password
23	RADIUS server shared secret	Used to authenticate the connection to a RADIUS server	MD5
24	TACACS+ server shared secret	Used to authenticate the connection to a TACACS+ server	MD5
Disk Encryption			
25	Encryption key	Used to encrypt Boost data pages on disk	AES-128-CBC

Item	Description/Usage	Description/Usage	Type
IKE-based IPsec (ECOS 8.1.9)			
26	IPSec PSK	Used for tunnel peer authentication	char string
27	IPSec Diffie-Hellman private key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
28	IPSec Diffie-Hellman public key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
29	IPSec Diffie-Hellman peer public key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
30	IPSec Diffie-Hellman shared secret	Used to derive encryption and HMAC keys	Diffie-Hellman algorithm
31	IPSec AES keys	Used during encryption and decryption of data within the IPSec protocol	AES-CBC (128, 256 bits)
32	IPSec HMAC keys	Used to protect integrity of data within the IPSec protocol	Using SHA1, SHA256, SHA384, SHA512
33	IPSec IKEv1 KDF internal state	Used to derive encryption and HMAC key from shared secret	SP800-135 IKEv1 KDF
IKE-based IPsec (ECOS 9.1)			
34	IPSec PSK	Used for tunnel peer authentication	char string
35	IPSec Diffie-Hellman private key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
36	IPSec Diffie-Hellman public key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
37	IPSec Diffie-Hellman peer public key	Used during the IPSec handshake to establish the shared secret	Diffie-Hellman Group 14-18 (default is DH group 14)
38	IPSec Diffie-Hellman shared secret	Used to derive encryption and HMAC keys	Diffie-Hellman algorithm
39	IPSec AES keys	Used during encryption and decryption of data within the IPSec protocol	AES-CBC (128, 256 bits)
40	IPSec HMAC keys	Used to protect integrity of data within the IPSec protocol	Using SHA1, SHA256, SHA384, SHA512
41	IPSec IKEv1 and IKEv2 KDF internal state	Used to derive encryption and HMAC key from shared secret	SP800-135 IKEv1 and IKEv2 KDF

Item	Description/Usage	Description/Usage	Type
ECOS Software Update			
42	Firmware update RSA key	Used to protect integrity during firmware update	RSA 4096 bits
SNMP			
43	SNMP Authentication and Privacy Password	Used to establish SNMP sessions	Password (20-32 characters)
44	SNMP Authentication and Privacy Secret	Used to establish SNMP sessions	SHA1, AES-128-CFB
45	SNMP KDF Internal States	Used to derive encryption and HMAC key from secret	SP800-135
SSHv2			
46	SSH Server RSA public key	Used to authenticate the SSH handshake	RSA 2048 bits
47	SSH Server RSA private key	Used to authenticate the SSH handshake	RSA 2048 bits
48	SSH Server Diffie-Hellman public key	Used during the SSH handshake to establish the shared secret	DH group 14: 2048 bits
49	SSH Server Diffie-Hellman private key	Used during the SSH handshake to establish the shared secret	DH group 14: 2048 bits
50	SSH Client Diffie-Hellman public key	Used during the SSH handshake to establish the shared secret	DH group 14: 2048 bits
51	SSH AES key	Used during encryption and decryption of data within the SSH protocol	AES-CTR (128 bits, 192 bits, 256 bits)
52	SSH HMAC key	Used to protect integrity of data within the SSH protocol	SHA1
53	SSH KDF internal states	Used to derive encryption and HMAC key from shared secret	Hash H
DRBG			
54	CTR_DRBG Entropy Input	Used during generation of random numbers	Intel RDRAND Entropy input length is equal to the AES key length (AES-128, 192, AES-256)

Item	Description/Usage	Description/Usage	Type
55	CTR_DRBG Seed	Used during generation of random numbers	NIST SP 800-90A CTR_DRBG (AES-128, 192, AES-256) with Derivation Function Seed length = key length + 128 bits
56	CTR_DRBG Internal states: V and Key	Used during generation of random numbers	NIST SP 800-90A CTR_DRBG (AES-128, 192, AES-256) with Derivation Function Value of V (128-bits) and Key (128, 192, and 256 bits)
End of Table			

Appendix A: Glossary of Terms

Table 4: Glossary

Abbreviation	Meaning
AES	Advanced Encryption Standard, as specified in [FIPS 197]
AES-GCM	AES with Galois/Counter Mode
ANSI	American National Standards Institute
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CMVP	Cryptographic Module Validation Program
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie-Hellman (Algorithm)
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman (NIST SP 800-56A)
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode (GCM) and GMAC Algorithm
HMAC	Keyed-Hash Message Authentication Code, as specified in [FIPS 198]
KAS	Key Agreement Schemes and Key Confirmation (NIST SP 800-56A)
KDF	Key Derivation Function
MAC	Message Authentication Code
MD5	Message Digest 5
NIST	National Institute of Standards and Technology
NRBG	Non-deterministic Random Bit Generator
PKCS	Public Key Cryptography Standard

Abbreviation	Meaning
PP	Protection Profile
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Cryptographic System (FIPS 186-4)
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
TLS	Transport Layer Security

Appendix B: References

https://csrc.nist.gov/publications/sp800

Table 5: References

Reference	Specification
FIPS 140-2	Security Requirements for Cryptographic modules, May 25, 2001
FIPS 180-4	Secure Hash Standard (SHS)
FIPS 186-4	Digital Signature Standard
FIPS 197	Advanced Encryption Standard (AES)
FIPS 198-1	The Keyed-Hash Message Authentication Code (HMAC)
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-800-52 Rev 2	Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations
SP 800-56A Rev 3	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 800-90A	Recommendation for Random Number Generation Using Deterministic Random Bit Generators

Reference	Specification
SP 800-108	Recommendation for Key Derivation Using Pseudorandom Functions
SP 800-132	Recommendation for Password-Based Key Derivation
SP 800-133 Rev 2	Recommendation for Cryptographic Key Generation
SP 800-135	Recommendation for Existing Application –Specific Key Derivation Functions