

MACsec Using EAP-TLS Authentication

This chapter describes how to achieve MACSec encryption between two Routers using the 802.1X port-based authentication with Extensible Authentication Protocol-Transport Layer Security (EAP-TLS).

For more information on 802.1X port-based authentication, see the *802.1 X Port-Based Authentication* chapter.

Table 1: Feature History

Release	Modification
Release 6.6.3	This feature was introduced.

- MACSec Using EAP-TLS Authentication, on page 1
- Configure MACSec Encryption Using EAP-TLS Authentication, on page 1
- PQC key exchange groups for EAP-TLS, on page 5

MACSec Using EAP-TLS Authentication

This chapter describes how to achieve MACSec encryption between two Routers using the 802.1X port-based authentication with Extensible Authentication Protocol-Transport Layer Security (EAP-TLS). EAP-TLS allows mutual authentication using certificates, between the authentication server and the client, and generates the Master Session Key (MSK). This MSK is used to derive the Connectivity Association Key (CAK), and the corresponding Connectivity Association Key Name (CKN) is derived from the EAP session ID.

The enable-tls1.3-legacy-kdf must be configured when using EAP based MACsec with Local EAP Authentication to ensure interoperability with XR releases earlier than Release 25.4.1.

Configure MACSec Encryption Using EAP-TLS Authentication

The system supports certificate-based MACsec encryption using both local and remote EAP-TLS authentications.

Restrictions for MACSec Using EAP-TLS Authentication

- The system does not support certificate-based (EAP-TLS) MACsec encryption on sub-interfaces.
- The system does not support MACSec using EAP-TLS authentication in **multi-auth** host mode.

You must also follow the guidelines and restrictions applicable to EAP-TLS session. For details, see the #unique_327 section in the *802.1X Port-Based Authentication* chapter.

Prerequisites

For MACSec using EAP-TLS authentication, you must first configure a EAP-TLS session. For more information on configuring EAP-TLS session, see the following topics in the *802.1X Port-Based Authentication* chapter:

- 802.1X with Remote RADIUS Authentication
- 802.1X with Local EAP Authentication
- Router as 802.1X Supplicant

The MKA participant with 802.1X PAE role as **authenticator** acts as the key server and the **supplicant** acts as the non-key server.

When the 802.1X PAE role for the interface is configured as **authenticator** or **both**, then you must configure the dot1x **host-mode** under the authenticator sub mode as **single-host** or **multi-host** in order to bring up the MACsec EAP session. For details, see 802.1X host-modes.

Configure MACSec EAP on an Interface

The following section describes the steps to configure MACSec EAP on an interface.

Configuration Example

```
Router#configure
Router(config)#interface HundredGigE 0/1/1/2
Router(config-if)#macsec eap
Router(config-if)#commit
```

Running Configuration

```
Router#show run interface HundredGigE 0/1/1/2
interface HundredGigE 0/1/1/2
    macsec eap
!
```

You can also configure MACSec EAP on an interface by specifying the configured MACSec policy name.

Configuration Example

```
Router(config-if)#macsec eap policy test-macsec-policy
```

Running Configuration

```
Router#show run interface HundredGigE 0/1/1/2
interface HundredGigE 0/1/1/2
    macsec eap policy test-macsec-policy
!
```

Configure the EAP-derived MACsec CAK length

Set the 128-bit Connectivity Association Key (CAK) length used for MACsec EAP sessions.

MACsec EAP supports configuring the EAP-derived CAK length as either 128 bits or 256 bits under a MACsec policy. The default EAP-derived CAK length is 128 bits.

Before you begin

Identify the MACsec policy used by the MACsec EAP session.

Follow these steps to configure the EAP-derived MACsec CAK length:

Procedure

Step 1 Configure the EAP-derived CAK length in the MACsec policy.

Example:

```
Router(config)# macsec-policy <policy-name> eap-cak-length 128
Router(config)# commit
```

Step 2 Display and verify the configured EAP-derived CAK length.

Example:

```
Router# show mka macsec policy <policy-name>
```

Changing the EAP-derived CAK length in a MACsec policy re-establishes MACsec sessions on all interfaces that use the policy with the newly configured CAK length.

What to do next

Configure the same EAP-derived CAK length in the MACsec policy on the peer.

Verify MACSec EAP Configuration on an Interface

You can use these commands to verify the MACSec EAP configuration:

- **show macsec mka session interface**

  Sample output:

```
Router# show macsec mka session interface HundredGigE 0/1/1/2
================================================================================
Interface-Name    Local-TxSCI     #Peers Status  Key-Server PSK/EAP CKN
================================================================================
Hu0/1/12      0201.9ab0.85af/0001    1   Secured YES        EAP     A94399 ...
```

- **show macsec mka session interface detail**

  Sample output:

```
Router# show macsec mka session interface HundredGigE 0/1/1/2 detail
```

```
MKA Detailed Status for MKA Session
===================================
Status                                  : SECURED - Secured MKA Session with MACsec

Local Tx-SCI                            : 0201.9ab0.85af/0001
Local Tx-SSCI                           : 2
Interface MAC Address                   : 0201.9ab0.85af
MKA Port Identifier                     : 1
Interface Name                          : Hu0/1/1/2
CAK Name (CKN)                          : A94399EE68B2A455F85527A4309485DA
CA Authentication Mode                  : EAP
Keychain                                : NA (EAP mode)
Member Identifier (MI)                  : 3222A4A7678A6BDA553FDB54
Message Number (MN)                     : 114
Authenticator                           : YES
Key Server                              : YES
MKA Cipher Suite                        : AES-128-CMAC
Configured MACSec Cipher Suite          : GCM-AES-XPN-256
Latest SAK Status                       : Rx & Tx
Latest SAK AN                           : 1
Latest SAK KI (KN)                      : 3222A4A7678A6BDA553FDB5400000001 (1)
Old SAK Status                          : No Rx, No Tx
Old SAK AN                              : 0
Old SAK KI (KN)                         : RETIRED (0)
SAK Transmit Wait Time                  : 0s (Not waiting for any peers to respond)
SAK Retire Time                         : 0s (No Old SAK to retire)
Time to SAK Rekey                       : NA
MKA Policy Name                         : *DEFAULT POLICY*
Key Server Priority                     : 16
Delay Protection                        : FALSE
Replay Window Size                      : 64
Include ICV Indicator                   : FALSE
Confidentiality Offset                  : 0
Algorithm Agility                       : 80C201
SAK Cipher Suite                        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability                       : 3 (MACsec Integrity, Confidentiality, &
Offset)
MACsec Desired                          : YES

# of MACsec Capable Live Peers          : 1
# of MACsec Capable Live Peers Responded : 1

Live Peer List:
MI                        MN     Rx-SCI (Peer)      SSCI    KS-Priority
-----------------------------------------------------------------------------
86B47DE76B42D9D7AB6805F7  113   0257.3fae.5cda/0001   1        16

Potential Peer List:
MI      MN         Rx-SCI (Peer)             SSCI         KS-Priority
-----------------------------------------------------------------------------

Peers Status:
Last Tx MKPDU   : 2018 Mar 01 13:36:56.450
Peer Count      : 1
RxSCI           : 02573FAE5CDA0001
MI              : 86B47DE76B42D9D7AB6805F7
Peer CAK        : Match
Latest Rx MKPDU : 2018 Mar 01 13:36:56.450
```

- **show macsec mka summary**

  Sample output:

```
Router#show macsec mka summary
```

```
================================================================================
Interface-Name Status   Cipher-Suite    KeyChain      PSK/EAP   CKN
================================================================================
Hu0/1/12       Secured  GCM-AES-XPN-256 NA(EAP mode)  EAP       A94399 ...

Total MACSec Sessions : 1
Secured Sessions : 1
Pending Sessions : 0
```

PQC key exchange groups for EAP-TLS

A PQC key exchange group is a TLS named group that

- combines classical and post-quantum cryptography (PQC) algorithms to protect key exchange,
- is advertised by IOS XR during 802.1X EAP-TLS negotiation, and
- enables support for hybrid, pure PQC, and classical key exchange methods for dynamic MACsec sessions.

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
PQC-enabled MACsec with EAP-TLS	Release 26.3.1	Introduced in this release on: NCS 5500 fixed port routers; NCS 5700 fixed port routers; NCS 5500 modular routers (NCS 5500 line cards; NCS 5700 line cards [Mode: Native]). Introduces PQC-capable TLS key exchange support to 802.1X EAP-TLS, with default CiscoSSL group advertisement and per-profile group configuration. This feature is supported on all MACsec-supported IOS XR platforms.

From IOS XR Release 26.3.1, EAP-TLS advertises CiscoSSL-supported PQC and classical groups by default when no explicit group list is configured. You can configure a colon-separated list on each EAP profile.

The feature supports the 802.1X EAP-TLS supplicant and the Local EAP authenticator. After successful authentication, the EAP-TLS session provides keying material for MACsec key establishment.

Note

- Hybrid groups are recommended during migration because they provide classical and PQC key exchange protection.
- On platforms that support port control, MACsec EAP does not support the `should-secure` MACsec security policy. Dot1x continues to perform port control as part of its default operation, which prevents `should-secure` from functioning with MACsec EAP.

Ensure compatibility and FIPS compliance for PQC EAP-TLS configuration

Follow these requirements when configuring PQC EAP-TLS key exchange groups:

- Ensure that the supplicant and authenticator have at least one key exchange group in common.
- If the peers have no group in common, EAP-TLS fails and the system logs an authentication or supplicant TLS failure with a no-common-group hint.
- Use only FIPS-approved groups when the system operates in FIPS mode.
- The system retains valid non-FIPS configured groups but excludes them from EAP-TLS authentication while FIPS mode is active.
- If FIPS excludes every configured group, the TLS session fails explicitly without using the default group list.
- Changing the configured key exchange groups restarts affected EAP-TLS and MACsec sessions.

Supported PQC and classical key exchange groups

Use this reference to select supported TLS key exchange groups for an EAP profile.

Table 3: Supported key exchange groups

Category	Groups	FIPS mode
Hybrid ML-KEM and ECDHE	x448_mlkem768, X25519MLKEM768, x25519_mlkem512, p256_mlkem512, p384_mlkem768, SecP256r1MLKEM768, p521_mlkem1024, SecP384r1MLKEM1024	p256_mlkem512, p384_mlkem768, SecP256r1MLKEM768, p521_mlkem1024, SecP384r1MLKEM1024
Pure ML-KEM	mlkem512, mlkem768, mlkem1024	Not permitted
Classical ECDHE	x25519, x448, secp256r1, secp384r1, secp521r1	secp256r1, secp384r1, secp521r1
Classical FFDHE	ffdhe2048, ffdhe3072, ffdhe4096	ffdhe2048, ffdhe3072, ffdhe4096

Configure PQC key exchange groups for EAP-TLS

Configure the key exchange groups that an EAP-TLS profile uses for dynamic MACsec authentication.

When no group list is configured, IOS XR uses the default group list provided by CiscoSSL. In FIPS mode, IOS XR considers only FIPS-approved groups during EAP-TLS negotiation. If a group list is configured, non-FIPS groups remain in the configuration but are not used while FIPS mode is active.

Note FIPS mode is enabled systemwide. If the router operates in FIPS mode, use only FIPS-approved key-exchange groups. For FIPS mode prerequisites, enablement, and operational guidance, refer to Configuring FIPS Mode.

Before you begin

- Identify the TLS trustpoint to be used.
- Ensure you have determined a group list that both EAP-TLS peers support.

Follow these steps to configure PQC key exchange groups for EAP-TLS:

Procedure

Step 1 Enter EAP profile TLS configuration mode.

Example:

```
Router(config)# eap profile <profile-name>
method tls
pki-trustpoint <trustpoint>
```

Step 2 Associate the EAP profile with the Dot1x profile.

Example:

```
Router(config)# dot1x
profile <dot1x-profile-name>
eap profile <eap-profile-name>
```

Step 3 Configure the colon-separated key exchange group list.

Example:

```
Router(config)# key-exchange-groups <colon-separated-list>
```

Step 4 To verify your configuration, display detailed Dot1x information for the interface to check TLS version, cipher suite, key exchange type, and group.

Example:

```
Router# show dot1x interface <interface-name> detail
```

Step 5 Display the configured groups in the EAP profile.

Example:

```
Router# show running-config eap profile <profile-name>
```

Changing the configured key exchange groups resets affected EAP-TLS and MACsec sessions and starts reauthentication.

What to do next

Configure the same or an overlapping group list on the EAP-TLS peer to ensure successful negotiation.

Troubleshoot EAP-TLS key exchange group negotiation

Identify and resolve mismatches in negotiated TLS key exchange groups causing EAP-TLS authentication failures.

When EAP-TLS authentication fails, IOS XR provides session-specific TLS failure hints in Dot1x output and may display FIPS warnings if FIPS mode excludes a configured key exchange group.

Before you begin

- Identify the affected interface.
- Collect the configured EAP profile and MACsec policy names.

Follow these steps to troubleshoot EAP-TLS key exchange group:

Procedure

Step 1 Check for `DOT1X-3-AUTH_TLS_FAILURE` or `DOT1X-3-SUPP_TLS_FAILURE` and their TLS failure hints. When FIPS excludes a configured non-FIPS group, check for `DOT1X-4-FIPS_WARN`.

Step 2 Display detailed Dot1x information for the affected interface.

Example:

```
Router# show dot1x interface <interface-name> detail
```

Check the TLS Fail Info field for a no-common-key-exchange-group hint or another session-specific TLS failure reason.

Step 3 Display the configured key exchange group list.

Example:

```
Router# show running-config eap profile <profile-name>
```

Compare the configured list with the group list on the peer and confirm that the lists overlap.

Step 4 Enable EAP method debugging when additional TLS handshake detail.

Example:

```
Router# debug eap method
```

The debug output reports ClientHello supported groups and versions, proposed cipher suites, and ServerHello selected values.

The diagnostic output identifies the TLS parameters that the peers proposed or selected and indicates whether configuration mismatch or FIPS filtering caused the failure.