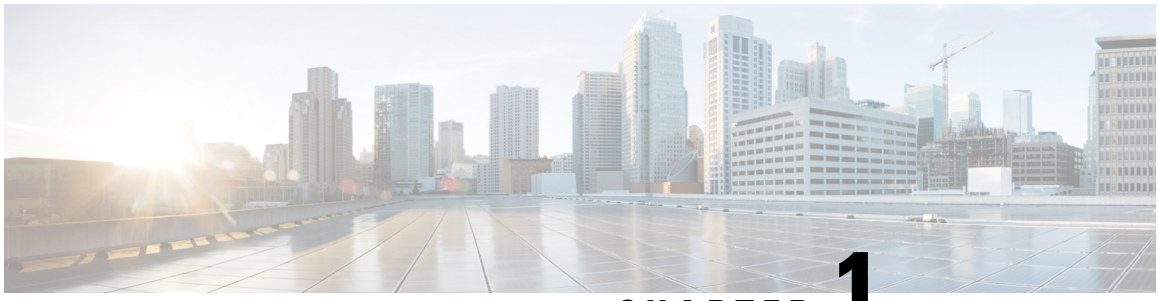
MACsec Configuration Guide for Cisco 8000 Series Routers, Cisco IOS XR Release

First Published: 2025-10-31

Last Modified: 2026-02-28

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

YANG data models for MACsec encryption features

This chapter provides information about the YANG data models for MACsec encryption features.

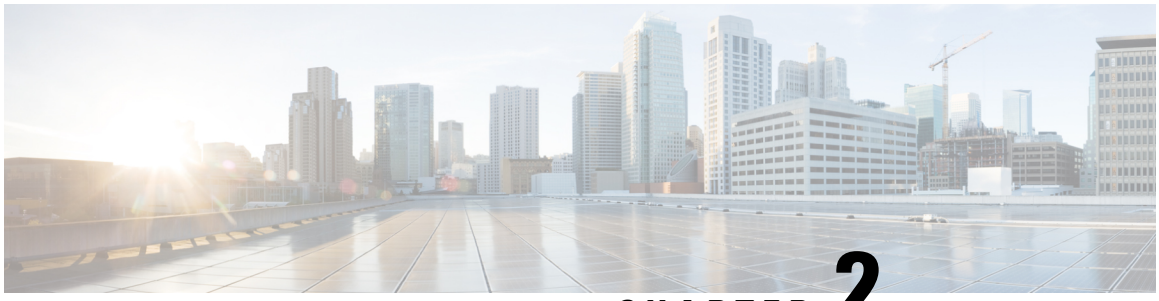
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the *Available-Content.md* file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 2

Fundamentals of MACsec encryption

This chapter provides a comprehensive overview of MACsec encryption fundamentals, including key concepts, deployment models, configuration steps, and verification procedures. Users can leverage this chapter to understand MACsec benefits, set up secure Layer 2 encryption on their routers, and follow best practices for configuration and key management.

MACsec encryption

MACsec encryption is a Layer 2 security technology that

- protects data on physical media from common attacks such as MAC address spoofing, ARP spoofing, Denial of Service (DoS) attacks targeting DHCP servers, and VLAN hopping
- provides data confidentiality and integrity by encrypting traffic at the physical layer,
- precedence over higher-layer encryption methods such as IPsec and SSL, and
- deploys on Customer Edge (CE) router interfaces that connect to Provider Edge (PE) routers and on all provider router interfaces.

Benefits of MACsec encryption

- Data integrity check: Uses an Integrity Check Value (ICV) sent with the protected data unit. The receiver recalculates and compares the ICV to detect any data modification.
- Data encryption: Enables a port to encrypt outbound frames and decrypt inbound frames encrypted with MACsec.
- Replay protection: Provides a configurable window that accepts a specified number of out-of-sequence frames to handle frames transmitted out of order.
- Support for clear traffic: Allows unencrypted data to transit through the port if configured accordingly.

Hardware support for MACsec encryption

The table lists the compatibility between specific Cisco IOS XR Software Releases and the corresponding hardware Product IDs (PIDs) that support MACsec encryption.

Table 1: Hardware support for MACsec encryption

Cisco IOS XR Software Release	Product ID (PID)
Release 25.4.1	• 8011-12G12X4Y-A • 8011-12G12X4Y-D • 8711-48Z-M
Release 25.3.1	8011-4G24Y4H-I
Release 25.1.1	8712-MOD-M
Release 24.4.1	8711-32FH-M
Release 24.3.1	• 8212-48FH-M • 88-LC1-52Y8H-EM
Release 7.10.1	Cisco 8608: • 86-MPA-14H2FH-M • 86-MPA-4FH-M • 86-MPA-24Z-M
Release 7.5.2	8202-32FH-M
Release 7.3.3	88-LC0-34H14FH
Release 7.3.15	88-LC0-36FH-M
Release 7.0.12	8800-LC-48H

MACsec encryption by interface type

- Physical interfaces (Standard MACsec): Applies security directly to a physical Ethernet port. This provides standard link-layer security within a LAN or between directly connected devices.
- L3 subinterfaces (WAN MACsec): Designed for service provider networks. It preserves the provider's outer VLAN tag in clear text while encrypting the customer's data payload. This allows the provider's network to switch traffic correctly and ensures end-to-end security.

Both physical interfaces and L3 subinterfaces support point-to-point (P2P) and point-to-multipoint (P2MP) MACsec encryption deployment models.

MACsec encryption deployment models

MACsec encryption supports two primary deployment models:

1. Point-to-Point (P2P): Secures a direct link between two endpoints.

2. Point-to-Multipoint (P2MP): Enables a single device to establish secure communications with multiple remote devices.

P2P MACsec encryption deployments

- LAN: Establishes secure Ethernet connectivity between two devices on the same local network.
- Over L2VPN (Pseudowire): Extends MACsec protection across a service provider network by encapsulating encrypted traffic over Layer 2 VPNs or pseudowires.

P2MP MACsec encryption deployments

- LAN: Establishes separate secure sessions from a central device to multiple peers on the same local network segment.
- Over VPLS (Virtual Private LAN Service):

  Establishes encrypted multipoint connectivity by creating a secure hub-and-spoke topology over a provider's VPLS network, connecting a central site with multiple branch locations.

P2P is suitable for securing direct links on LANs and across service provider networks. P2MP is ideal when a single device must securely communicate with multiple endpoints, especially in hub-and-spoke topologies over VPLS. Both deployment models, P2P and P2MP MACsec encryption, are supported on physical interfaces and L3 subinterfaces.

Key concepts for MACsec encryption

MACsec Key Agreement protocol

MACsec Key Agreement (MKA) is a protocol that manages the secure exchange of cryptographic keys for MACsec. It establishes and maintains secure associations between devices, enabling encrypted communication over Ethernet links. MKA handles key distribution, authentication, and rekeying processes to ensure continuous data confidentiality and integrity.

MACsec Pre-shared Key

MACsec Pre-shared Key (PSK) is a static key shared between devices before communication begins. It serves as a basis for authenticating devices and deriving session keys in MACsec. PSK simplifies deployment in environments where dynamic key management is not feasible but requires secure key distribution and management practices.

- Connectivity Association Key Name (CKN): CKN is an identifier used to associate devices within a MACsec connectivity association. It uniquely identifies the keying material group and helps devices recognize peers that share the same security context. CKN ensures that only authorized devices participate in the secure communication.
- Connectivity Association Key (CAK): CAK is the primary cryptographic key shared among devices in a MACsec connectivity association. It is used to derive session keys for encrypting and authenticating data frames. CAK must be securely distributed and protected to maintain the integrity and confidentiality of the MACsec session.

Fallback PSK and active fallback

Fallback PSK is a session recovery mechanism that activates when the primary PSK fails to establish a secured MKA session, ensuring a PSK is always available for MACsec encryption and decryption. Cisco IOS XR software enhances fallback PSK with the active fallback, which initiates a fallback MKA session when fallback configuration is present on the interface. Active fallback ensures faster session convergence on fallback during primary key deletion, expiry, or mismatch. It also accelerates traffic recovery under the should-secure security policy when both primary and fallback keys mismatch.

Secure Association Key

The actual encryption key that the key server generates and distributes to the key client. Each secure channel uses a new Secure Association Key (SAK) for data encryption.

- Key server: A router selected during the MKA process that is responsible for generating and distributing the SAK. Its selection is based on configured priority values, where a numerically lower value indicates higher preference.
- Key client: The peer router that receives the SAK from the key server.

MACsec frame format

The MACsec frame format defines the structure of a frame after Media Access Control Security (MACsec) encryption. It consists of specific components that ensure data confidentiality, integrity, and authenticity at Layer 2.

Figure 1: MACsec frame format

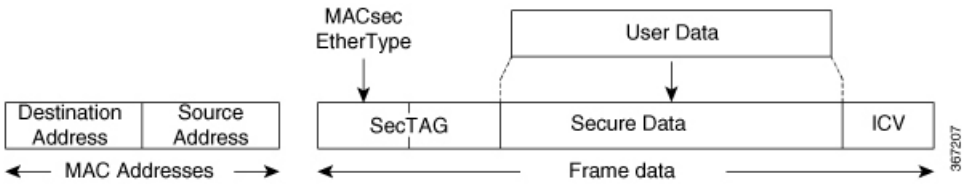


Table 2: MACsec frame components

MACsec frame component	What it is	Used for
SecTAG	A security tag, 8 to 16 bytes in length (16 bytes if Secure Channel Identifier (SCI) encoding is used, otherwise 8 bytes). It also provides replay protection.	Identifying the Secure Association Key (SAK) used for the frame and detecting out-of-sequence frames.
Secure Data	The portion of the frame containing data encrypted using MACsec, with a length of 2 or more octets.	Carrying encrypted data within the frame.
ICV (Integrity Check Value)	A value that provides an integrity check for the entire frame, typically ranging from 8 to 16 bytes in length.	Ensuring the integrity of the frame; frames with an ICV that does not match the expected value are dropped at the receiving port.

MACsec keychain

A MACsec keychain is a collection of cryptographic keys used to authenticate peers that need to exchange encrypted information. It defines the keys, their associated key strings (passwords), the cryptographic algorithm to be used, and the validity period for each key.

Table 3: MACsec keychain elements

MACsec keychain element	What it is	Used for
Key (CKN)	An identifier for the MACsec secret key.	Identifying each key entry in a MACsec keychain.
Key-string (CAK)	The actual secret key in the MACsec encryption.	Encrypting data based on the cryptographic algorithm used.
Cryptographic Algorithm	Specifies the encryption algorithm.	Determining how the key-string (CAK) is used for encryption.
Lifetime	Defines the validity period of the key, either as a duration or indefinitely.	Ensuring the key is used only within its valid time frame for security purposes.

MACsec policy

A MACsec policy defines the security parameters and behaviors for Media Access Control Security (MACsec) encryption in routers. It specifies the cryptographic algorithms, key management preferences, and traffic handling rules for secure Layer 2 communication.

MACsec policy encompasses several key parameters that govern MACsec operation:

Table 4: MACsec policy parameters

MACsec policy parameter	What it is	What it does
Cipher Suite	The encryption algorithm used for MACsec.	Provides the cryptographic strength and method for MACsec data encryption.
Confidentiality Offset	An offset value for MACsec encryption.	Modifies the starting point of encryption within a frame. Changes are recommended only when the port is administratively down to prevent traffic loss.
Key Server Priority	A value that determines a router's preference to be selected as the key server in an MKA session. A numerically lower value indicates higher preference.	Influences which router becomes the key server, responsible for generating and maintaining the Secure Association Key (SAK).
Security Policy	Defines the traffic handling behavior based on MACsec encryption status.	Controls whether unencrypted traffic is allowed before the MKA session secures, or if only encrypted traffic is permitted.
Data Delay Protection	A feature that ensures MACsec-protected data frames do not exceed a specific delay threshold.	Rejects MACsec-protected traffic that experiences excessive delay (over 2 seconds) to maintain real-time performance.

MACsec policy parameter	What it is	What it does
Replay Protection Window Size	The maximum number of out-of-sequence frames that are accepted.	Protects against replay attacks by defining the acceptable window for frame reordering.
Include ICV Indicator	A configuration option for including an optional Integrity Check Value (ICV) Indicator in the transmitted MACsec Key Agreement PDU (MKPDU).	Ensures interoperability with other vendor MACsec implementations that expect this specific indicator in the MKPDU.
SAK Rekey Interval	A timer value for periodically rekeying the MACsec Secure Association Key (SAK).	Periodically updates the data encryption key (SAK) to enhance security by limiting the lifespan of a single key. This configuration is effective on the node acting as the key server.

How MACsec encryption works

MACsec is a Layer 2 IEEE 802.1AE standard that secures data on physical media by encrypting packets between two MACsec routers.

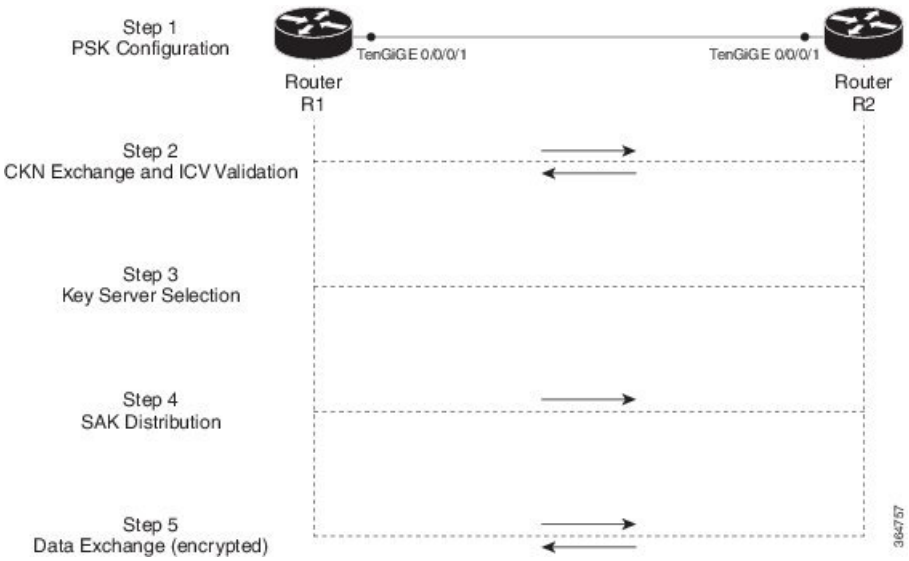
Summary

The key components involved in MACsec encryption are:

- MACsec routers: Devices that implement the MACsec standard to encrypt and decrypt traffic.
- MACsec Key Agreement (MKA) protocol: Manages the exchange of session keys and encryption keys.
- Pre-shared Key (PSK): A shared secret used for mutual peer authentication.
- Secure Association Key (SAK): The actual encryption key used for data encryption.
- MACsec frame format: The structure of encrypted packets, including SecTAG, Secure Data, and ICV.

Workflow

Figure 2: MACsec encryption process



These stages describe how MACsec encryption works:

1. Link establishment and peer authentication: When two MACsec routers first connect, they establish a peer relationship. Both devices perform mutual authentication using a pre-shared key (PSK).
2. Connectivity association formation: After successful peer authentication, the routers create a connectivity association. They exchange a secure connectivity association key name (CKN) and validate the media key agreement (MKA) integrity check value (ICV) using the connectivity association key (CAK).
3. Key server selection: The routers select a key server based on their configured priorities.

   Rules that apply to key server selection include:

   - Lower numerical values of key server priority and SCI receive the highest preference.
   - A lower priority value increases the preference for the router to become the key server, while the other router functions as a key client. If no value is configured, the default value of 16 is taken to be the key server priority value for the router.
   - Each router selects a peer advertising the highest preference as its key server if peer has not selected another router as its key server or is not willing to function as the key server.
   - If two routers tie for the highest preference, a router with the highest priority SCI becomes the key server (KS).

4. Security association and SAK distribution: The selected key server generates and distributes the secure association key (SAK). Each secure channel relies on a series of overlapping security associations (SA), with each SA utilizing a new SAK.
5. Encrypted data exchange: Once the routers distribute the SAKs and establish security associations, they begin exchanging encrypted data. The data frames include a MACsec header with a SecTAG (for SAK identification and replay protection), the secure data (the encrypted payload), and an ICV (for integrity checking). Once assembled, both devices transmit the encrypted data.

Result

The MACsec process secures data on physical media, making it impossible for data to be compromised at higher layers. It provides data integrity checks, data encryption, and replay protection. This enhances the overall security of the network.

Guidelines for MACsec encryption

To ensure secure and reliable MACsec encryption:

- Use strong keychains to protect MACsec credentials and keys.
- Safely manage fallback preshared keys (PSK) and configure active fallback settings to support redundancy.
- Consistently apply MACsec encryption configuration to all relevant interfaces to prevent security gaps.

Guidelines for configuring MACsec keychains

Follow these guidelines to effectively and securely manage MACsec keychains:

- Ensure that the MACsec Connectivity Association Key Name (CKN) and Connectivity Association Key (CAK) match exactly on both ends. If the CKN or CAK do not match, the MKA session cannot be established, resulting in failed secure communication.
- Use unique, case-insensitive key IDs for each MACsec key to prevent session instability. MACsec key IDs are case-insensitive and stored in uppercase (for example, 'FF' and 'ff' are treated the same), so duplicate IDs may cause session instability. This case insensitivity does not apply to Netconf protocol configurations.
- Use MACsec keys of even length, up to 64 characters. Odd-length keys cause the system to exit MACsec configuration mode, preventing key setup.
- Always use the latest key in the keychain for MKA protocol operations. The key with the most recent Start Time among active keys is automatically used. You can verify key details with the **show key chain** command.
- Activate new MACsec keys in advance to ensure at least a one-minute overlap with the current key, ensuring seamless CAK rollover and preventing session interruptions.
- Set Start and Expiry times with future timestamps to automate CAK rotation. Automating key rotation enables bulk configuration for daily CAK rotation without manual intervention, improving operational efficiency and security.
- Do not delete or allow the current active key to expire. Deleting or allowing the active key to expire will terminate the MKA session and disrupt traffic. To prevent service interruption, configure keys with an infinite lifetime. If fallback is enabled, traffic will continue by switching to the fallback key upon expiry or deletion of the primary active key.
- Monitor key status regularly and take action before a key expires.When a key expires, the MACsec session terminates and secure connectivity is lost. Use the following commands to check status:
  - **show macsec mka session**: Displays no session information if key expires.

- **show macsec mka interface detail**: Displays `*** No Active Keys Present ***` in the PSK information.

Guidelines for managing fallback PSK and active fallback

Follow these guidelines to ensure seamless and secure key management during MACsec operations:

- Ensure the system performs a hitless rollover from the current active key to the fallback key during CAK rollover of primary keys if the latest active keys mismatch and the fallback keys match.
- Ensure the system performs a hitless rollover back to the primary latest active key when a session is active with the fallback key and the primary latest active key mismatch is resolved between peers.
- Enable active fallback to include the fallback PSK entry in MACsec show commands. When the session is secured with the primary key, the fallback session status must display as ACTIVE.
- Configure a valid fallback PSK (CKN and CAK) with an infinite lifetime.
- Do not configure the fallback PSK with a CAK mismatch. If a mismatch happens, resolve it by pushing a new set of PSK configurations across all association members—first on the fallback PSK keychain, then on the primary PSK keychain.
- Configure the enable-legacy-fallback command under the macsec-policy to maintain backward compatibility if the peer device runs an older software release that does not support active fallback.
- In point-to-point (P2P) topologies, rollover to the fallback PSK occurs when either node in the Secure Association (SA) cannot establish a session with the primary PSK.
- In point-to-multipoint (P2MP) topologies, fallback occurs only when the primary key expires or is deleted on all peers, not just one. If the primary PSK is deleted or expires on a single node (e.g., R1), a new key server is selected among the remaining peers to perform a SAK rekey. This process excludes that node from the SA. All traffic to and from that node is dropped.

Guidelines to configure MACsec interface

Follow these guidelines to ensure optimal configuration and performance of MACsec interfaces:

- Configure separate keychains for primary and fallback PSKs. Do not update both PSKs at the same time. Use the fallback PSK only to recover a MACsec session if the primary key fails.
- Adjust the interface MTU to account for MACsec overhead. For example, if the default MTU is 1514 bytes, set it to 1546 bytes (1514 + 32 bytes overhead). For IS-IS, ensure a minimum MTU of 1546 bytes.
- Enable MACsec on all members of a bundle.
  - If MACsec peers use IOS-XR version 24.1.1 or higher, configure **impose-overhead-on-bundle** in the MACsec policy to adjust the bundle interface MTU for routing protocols running on the bundle interface.
  - If using IOS-XR versions prior to 24.1.1, configure the maximum MTU on the bundle interface to accommodate the protocol packet size plus 32 bytes MACsec overhead. Disable hello-padding for IS-IS running on the bundle interface.

- Define the MACsec keychain before applying the MACsec configuration to the interface. If you apply the keychain without specifying a policy, the default MACsec policy is used.
- Use the `openconfig-macsec.yang` OpenConfig data model to programmatically view the MACsec configuration. For more information, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Configure MACsec encryption

Configure MACsec encryption on network interfaces to secure data.

To enable secure communication on physical interfaces, configure MACsec encryption with specific settings.

Procedure

Step 1 Create a MACsec keychain to manage encryption keys.

Step 2 Define a user-defined MACsec policy to specify security requirements.

Step 3 Apply the MACsec configuration to physical interfaces.

MACsec encryption is successfully configured on the specified physical interfaces.

Configure a MACsec keychain

Enable MACsec encryption and decryption on routers by configuring a MACsec keychain, ensuring secure communication between peers using the MACsec protocol.

Follow these steps to configure a MACsec keychain:

Before you begin

- Ensure you have administrative access to the router.
- Verify that the router supports MACsec encryption.

Procedure

Step 1 Create a keychain, specifying a unique keychain name.

Example:

```
Router# configure
Router(config)# key chain kc
```

Step 2 Enable MACsec mode for the keychain.

Example:

```
Router(config-kc)# macsec
```

Step 3 Configure a MACsec key for the keychain.

Example:

```
Router(config-kc-MacSec)# key key1
```

Step 4 Specify the key string and the cryptographic algorithm.

Example:

```
Router(config-kc-MacSec-KEY1)# key-string
112233445566778899900AABBCCDDEEFF00112233445566778899AABBCCDDEEFF cryptographic-algorithm AES-128-CMAC-96
```

- Key-string range: The key-string range is 32 characters for AES-128 and 64 characters for AES-256. Ensure that the string length matches the requirements of the selected algorithm.
- Cryptographic algorithm options: AES-128-CMAC-96 or AES-256-CMAC.

Step 5 Define the validity period for the key.

Example:

```
Router(config-kc-MacSec-KEY1)# lifetime 05:00:00 01 January 2019 infinite
Router(config-kc-MacSec-KEY1)# commit
```

Lifetime range: You can specify a lifetime range by providing a fixed timeframe (including start and expiry), or set it as infinite.

Step 6 Verify the keychain settings in the running configuration.

Example:

```
Router# show running-config key chain kc1
key chain kc1
 macsec
  key key1
    key-string 112233445566778899900AABBCCDDEEFF00112233445566778899AABBCCDDEEFF cryptographic-algorithm
 AES-128-CMAC-96
     lifetime 05:00:00 01 January 2019 infinite
  !
 !
```

The MACsec keychain is created and ready for use with MACsec encryption.

What to do next

Apply the keychain to the router interface configuration when required.

Create a user-defined MACsec policy

Define and configure a custom MACsec policy to secure network traffic. Specify encryption, key server priority, security parameters, and additional protections.

Follow these steps to create a user-defined MACsec policy:

Before you begin

- Ensure you have administrative access to the router.

- Verify that the router supports MACsec encryption.

Procedure

Step 1 Create a MACsec policy, specifying a unique policy name.

Example:

```
Router# configure
Router(config)# macsec-policy mp1
```

Step 2 Configure the cipher suite for MACsec encryption.

Example:

```
Router(config-macsec-policy)# cipher-suite GCM-AES-XPN-128
```

The GCM encryption method, which uses the AES encryption algorithm, supports the following encryption suites:

- GCM-AES-XPN-128
- GCM-AES-XPN-256

Step 3 Set the confidentiality offset value.

Example:

```
Router(config-macsec-policy)# conf-offset CONF-OFFSET-30
```

Step 4 Configure the key server priority.

Example:

```
Router(config-macsec-policy)# key-server-priority 10
```

Range: `0` to `255` (A lower value indicates higher priority for key server selection. Default value is `16`).

Step 5 Set the security policy:

Example:

```
Router(config-macsec-policy)# security-policy should-secure
```

- `must-secure`: Allows only MACsec-encrypted traffic. The router drops traffic until the MKA session is secured.
- `should-secure`: Allows unencrypted traffic until the MKA session is secured, then only encrypted traffic is allowed.

Step 6 Enable data delay protection.

Example:

```
Router(config-macsec-policy)# delay-protection
```

Step 7 Configure the replay protection window size.

Example:

```
Router(config-macsec-policy)# window-size 64
```

Range: `0` to `1024`

Step 8 Include the Integrity Check Value (ICV) indicator in frames that arrive on the port and commit the configuration to save the MACsec policy settings.

Example:

```
Router(config-macsec-policy)# include-icv-indicator
Router(config-macsec-policy)# commit
```

To set the rekey interval, use the **sak-rekey-interval** command in macsec-policy configuration mode. The timer ranges from 60 to 2,592,000 seconds, the default being OFF.

Step 9 Verify the MACsec policy settings in the running configuration.

Example:

```
Router# show running-config macsec-policy mp1

macsec-policy mp1
 conf-offset CONF-OFFSET-30
 security-policy should-secure
 cipher-suite GCM-AES-XPN-128
 window-size 64
 include-icv-indicator
 delay-protection
 key-server-priority 10
!
```

The user-defined MACsec policy is created and ready for use with MACsec encryption.

What to do next

Apply the user-defined MACsec policy to the router interface configuration when required.

Configure MACsec encryption on an interface

Secure network communication on a host-facing interface using MACsec encryption.

he MACsec PSK (keychain and user-defined policy) configuration is applied to a host-facing interface of a CE router. This establishes a secure connection.

Follow these steps to configure MACsec on an interface:

Before you begin

Ensure the interface is a host-facing interface on a CE router.

Procedure

Step 1 Access interface configuration mode.

Example:

```
Router# configure
Router(config)# interface hundredGigE Hu0/1/0/10
```

Step 2 Configure the IPv4 address for the interface.

Example:

```
Router(config-if)# ipv4 address 192.168.30.1 255.255.255.0
```

Step 3 Apply the MACsec keychain and user-defined MACsec policy to the interface.

Example:

```
Router(config-if)# macsec psk-keychain kc1 policy mp1
```

Step 4 Commit the configuration to save changes.

Example:

```
Router(config-if)# commit
```

Step 5 Verify the MACsec configuration applied to the interface.

Example:

```
Router# show running-config interface HundredGigE 0/1/0/10
interface HundredGigE 0/1/0/10
 ipv4 address 192.168.30.1 255.255.255.0
 macsec psk-keychain kc1 policy mp1
!
```

MACsec encryption is applied to the specified interface, securing communication.

Verify MACsec session status

Confirm that MACsec encryption is correctly configured and operational on your network devices.

After configuring MACsec on your routers, perform this task to ensure security and connectivity.

Follow these steps to verify MACsec encryption:

Before you begin

- Ensure MACsec is configured on the relevant interfaces.
- Access the executive mode on your router.

Procedure

Step 1 Verify the MACsec policy configuration using the **show macsec policy detail** command.

Example:

```
Router# show macsec policy mp1 detail
Policy Name                 : mp1
      Cipher Suite          : GCM-AES-XPN-128
      Key-Server Priority   : 10
      Window Size           : 64
      Conf Offset           : 30
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Should Secure
      Vlan Tags In Clear    : 1
```

```
LACP In Clear         : FALSE
LLDP In Clear         : FALSE
Pause Frame In Clear  : FALSE
Sak Rekey Interval    : FALSE
Include ICV Indicator : TRUE
Use Eapol PAE in ICV  : FALSE
Disable Suspend On Request    : FALSE
Disable Suspend For           : FALSE
Enable legacy fallback        : FALSE
SKS Profile                   : N/A
Max AN                        : 3
```

If the displayed values do not match your expected settings, run the **show run macsec-policy** command review your configuration.

Step 2 View summary of the MACsec sessions using the **show macsec mka summary** command.

Example:

```
Router# show macsec mka summary

NODE: node0_1_CPU0
================================================================================================
Interface-Name        Status  Cipher-Suite       KeyChain   PSK/EAP   CKN
================================================================================================
Hu0/1/0/10            Secured GCM-AES-XPN-128    kc         PRIMARY   1234

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

Step 3 Verify interface peering using the **show macsec mka session** command.

Example:

```
Router# show macsec mka session

NODE: node0_1_CPU0
=============================================================================================================
Interface-Name        Local-TxSCI           #Peers Status  Key-Server PSK/EAP CKN
=============================================================================================================
Hu0/1/0/10            7872.5d1a.e7d4/0001   1      Secured NO         PRIMARY 1234
```

Step 4 View details of the MKA session using the **show macsec mka session detail** command.

Example:

```
Router# show macsec mka session detail

NODE: node0_1_CPU0

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

Local Tx-SCI                    : 7872.5d1a.e7d4/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : 7872.5d1a.e7d4
MKA Port Identifier             : 1
Interface Name                  : Hu0/1/0/10
CAK Name (CKN)                  : 1234
CA Authentication Mode          : PRIMARY-PSK
Keychain Member Identifier (MI) : kc
Message Number (MN)             : C12A70FEE1212B835BDDDCBA
Authenticator                   : 3009
Key Server                      : NO
```

```
MKA Cipher Suite                 : NO : AES-128-CMAC
Configured MACSec Cipher Suite   : GCM-AES-XPN-128

Latest SAK Status                : Rx & Tx
Latest SAK AN                    : 0
Latest SAK KI (KN)               : 018E2F0D63FF2ED6A5BF270E00000001 (1)
Old SAK Status                   : FIRST-SAK
Old SAK AN                       : 0
Old SAK KI (KN)                  : FIRST-SAK (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
Time to exit suspension          : NA

MKA Policy Name                  : mp-SF
Key Server Priority              : 10
Delay Protection                 : TRUE
Replay Window Size               : 64
Include ICV Indicator            : TRUE
Confidentiality Offset           : 30
Algorithm Agility                : 80C201
SAK Cipher Suite                 : 0080C20001000003 (GCM-AES-XPN-128)
MACsec Capability                : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired                   : YES

# of MACsec Capable Live Peers   : 1
# of MACsec Capable Live Peers Responded : 0

Live Peer List:
-----------------------------------------------------------------------------
SSCI               MI                  MN   Rx-SCI               KS-Priority
-----------------------------------------------------------------------------
018E2F0D63FF2ED6A5BF270E 2699 008a.962d.7400/0001 2 16

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

Peers Status:
Last Tx MKPDU                    : 2019 Oct 08 09:07:06.475
Peer Count                       : 1

RxSCI                            : 008A962D74000001
MI                               : 018E2F0D63FF2ED6A5BF270E
Peer CAK                         : Match
Latest Rx MKPDU                  : 2019 Oct 08 09:07:06.032
```

Step 5 View detailed MKA session information for a specific interface using the **show macsec mka session interface** command.

Example:

```
Router# show macsec mka session interface hundredGigE 0/1/0/10

===============================================================================================
Interface-Name        Local-TxSCI           #Peers Status Key-Server PSK/EAP CKN
===============================================================================================
Hu0/1/0/10            7872.5d1a.e7d4/0001   1      Secured NO         PRIMARY 1234
Hu0/1/0/10            7872.5d1a.e7d4/0001   1      Secured NO         FALLBACK 5678
```

The `Status` field should indicate `Secured` for the MKA session. A status of `Pending` or `INITIALIZING` means MACsec encryption is not successfully configured.

Step 6 Verify MACsec session counter statistics using the **show macsec mka statistics** command.

Example:

```
Router# show macsec mka statistics interface hundredGigE 0/1/0/10

MKA Statistics for Session on interface (Hu0/1/0/10)
=====================================================
Reauthentication Attempts.. 0

CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated.... 0
Group CAKs Received..... 0

SA Statistics
SAKs Generated.......... 0
SAKs Rekeyed............ 0
SAKs Received........... 1
SAK Responses Received.. 0

MKPDU Statistics
MKPDUs Transmitted...... 3097
"Distributed SAK".. 0
"Distributed CAK".. 0
MKPDUs Validated & Rx... 2788
"Distributed SAK".. 1
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success.......... 3097
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUS No Tx on intf down.. 3
MKPDUS No Rx on intf down.. 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error............ 0
MKPDUs Rx Success.......... 2788
MKPDUs Rx Invalid Length... 0
MKPDUs Rx Invalid CKN...... 0
MKPDUs Rx force suspended.. 0
MKPDUs Tx force suspended.. 0

MKPDU Failures
MKPDU Rx Validation (ICV).................. 0
MKPDU Rx Bad Peer MN....................... 0
MKPDU Rx Non-recent Peerlist MN............ 0
MKPDU Rx Drop SAKUSE, KN mismatch......... 0
MKPDU Rx Drop SAKUSE, Rx Not Set.......... 0
MKPDU Rx Drop SAKUSE, Key MI mismatch..... 0
MKPDU Rx Drop SAKUSE, AN Not in Use....... 0
MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set.... 0
MKPDU Rx Drop Packet, Ethertype Mismatch.. 0
MKPDU Rx Drop Packet, Source MAC NULL..... 0
MKPDU Rx Drop Packet, Destination MAC NULL 0
MKPDU Rx Drop Packet, Payload NULL........ 0

SAK Failures
SAK Generation................... 0
Hash Key Generation.............. 0
SAK Encryption/Wrap.............. 0
SAK Decryption/Unwrap............ 0

CA Failures
```

```
ICK Derivation.................. 0
KEK Derivation.................. 0
Invalid Peer MACsec Capability... 0

MACsec Failures
Rx SC Creation.................. 0
Tx SC Creation.................. 0
Rx SA Installation.............. 0
Tx SA Installation.............. 0
```

Review the counters for MACsec PDUs transmitted, validated, and received, and check for transmission errors.

Step 7 Verify MACsec encryption and hardware interface descriptor block (IDB) information using the **show macsec ea idb interface** command.

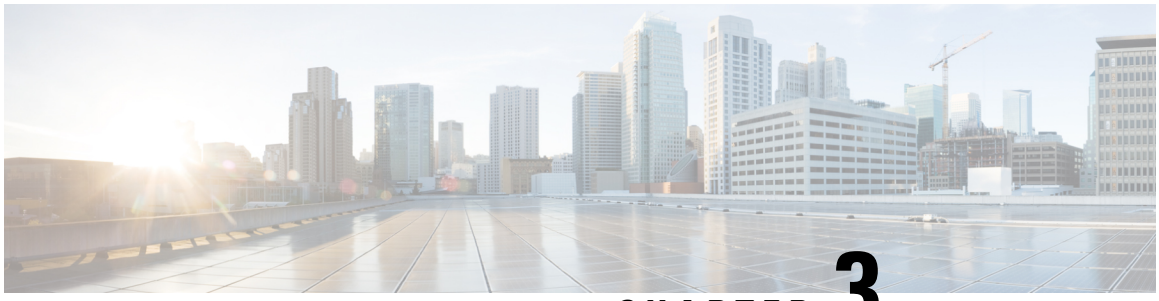
Example:

```
Router# show macsec ea idb interface hundredGigE 0/1/0/10

IDB Details:
if_sname                    : Hu0/1/0/10
if_handle                   : 0x8001e0
MacSecControlledIfh         : 0x800330
MacSecUnControlledIfh       : 0x800338
Replay window size          : 64
Local MAC                   : 78:72:5d:1a:e7:d4
Rx SC Option(s)             : Validate-Frames Replay-Protect
Tx SC Option(s)             : Protect-Frames Always-Include-SCI
Security Policy             : SHOULD SECURE
Delay Protection            : TRUE
Sectag offset               : 0

Rx SC 1
Rx SCI                      : 008a962d74000001
Peer MAC                    : 00:8a:96:2d:74:00
Stale SAK Data              : NO
SAK[0]                      : ***
SAK Len                     : 16
SAK Version                 : 1
HashKey[0]                  : ***
HashKey Len                 : 16
Conf offset                 : 30
Cipher Suite                : GCM-AES-XPN-128
CtxSalt[0]                  : 01 8f 2f 0f 63 ff 2e d6 a5 bf 27 0e
ssci                        : 2
Rx SA Program Req[0]: 2019 Oct 08 07:37:14.870
Rx SA Program Rsp[0]: 2019 Oct 08 07:37:14.902

Tx SC
Tx SCI                      : 78725d1ae7d40001
Active AN                   : 0
Old AN                      : 255
Next PN                     : 1, 0, 0, 0
SAK Data                    : ***
SAK[0]                      : 16
SAK Len                     : 1
SAK Version                 : ***
HashKey[0]                  : 16
HashKey Len                 : 30
Conf offset                 : GCM-AES-XPN-128
Cipher Suite                : 01 8f 2f 0c 63 ff 2e d6 a5 bf 27 0e
CtxSalt[0]                  : 1
ssci                        : 2019 Oct 08 07:37:14.908
Tx SA Program Req[0]: 2019 Oct 08 07:37:14.931
Tx SA Program Rsp[0]: 2019 Oct 08 07:37:14.931
```

Step 8 Verify hardware programming using the **show macsec platform hardware sa interface** command.

Example:

```
Router# show macsec platform hardware sa interface hundredGigE 0/1/0/10
----------------------------------------------------------------
Tx SA Details:
----------------------------------------------------------------
SCI : 7872.5d1a.e7d4/0001
Crypto Algo : GCM-AES-XPN-128
AES Key Len : 128 bits
AN : 0
Initial Packet Number : 1
Current Packet Number : 1
Maximum Packet Number : 3221225400
XForm in Use : YES
Action Type : SA Action Egress
Direction : Egress
Conf Offset : 00000030
Drop Type : 0x00000003
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
----------------------------------------------------------------
Rx SA Details:
----------------------------------------------------------------
SCI : 008a.962d.7400/0001
Replay Window : 64
Crypto Algo : GCM-AES-XPN-128
AES Key Len : 128 bits
AN : 0
Initial Packet Number : 1
```



CHAPTER 3

WAN MACsec encryption

This chapter provides comprehensive guidance on deploying and configuring MACsec encryption for secure Ethernet encryption across WAN environments. Users can learn how to apply MACsec on physical interfaces and Layer 3 subinterfaces, set VLAN-based policies, and customize EAPoL Ether-types and destination addresses to enhance security and interoperability in diverse network topologies.

WAN MACsec encryption

WAN MACsec encryption is a solution that

- provides end-to-end encryption across Layer 2 Ethernet WAN services,
- supports both point-to-point (P2P) and point-to-multipoint (P2MP) topologies, and
- is based on the IEEE 802.1AE standard for MACsec, which offers hop-by-hop encryption at the data link layer.

Use WAN MACsec to protect Ethernet frames with confidentiality, integrity, and origin authentication. You can extend traditional MACsec LAN encryption to WAN environments to achieve robust, standards-based, high-speed encryption across Ethernet WAN services. WAN MACsec helps you secure your data in transit across various WAN topologies while maintaining flexibility, performance, and interoperability.

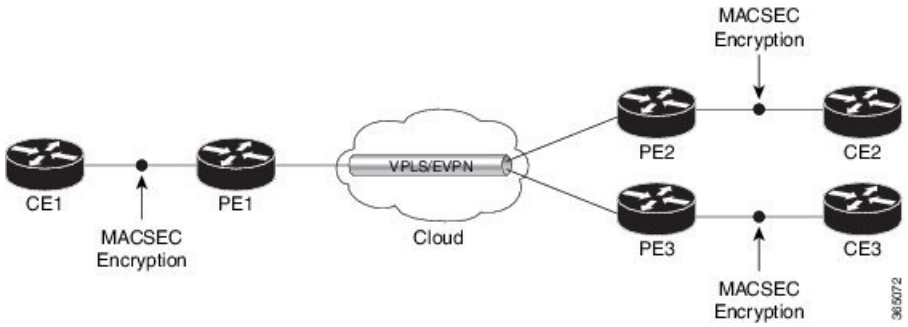
Applications of MACsec in WAN environments

To elucidate the application of MACsec in Wide Area Network (WAN) environments, with a specific emphasis on its implementation in VPLS/EVPN networks and MPLS core networks. This section outlines the configuration of MACsec on physical interfaces and link bundles to improve data security between geographically distributed data centers.

Use Case 1: MACsec in a VPLS/EVPN

In a typical Virtual Private LAN Service (VPLS) network, the risk of labeled traffic injection by potential hackers is prevalent. To counter this, MACsec is implemented in a VPLS/EVPN network to encrypt data exchanged over the VPLS cloud. In this topology, MACsec is configured on the provider edge (PE)-facing interfaces of the customer edge (CE) routers.

Figure 3: MACSec in a VPLS/EVPN Cloud

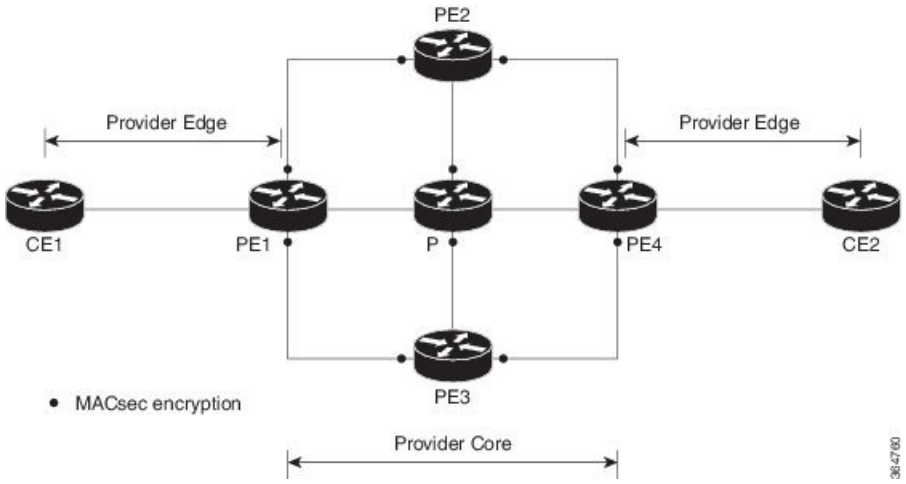


Use Case 2: MACsec in an MPLS Core Network

MACsec can be deployed in a Multiprotocol Label Switching (MPLS) core network on either physical interfaces or link bundles, also known as Link Aggregation Groups (LAG). This setup is particularly beneficial for MPLS networks that connect data centers located in different geographies, ensuring that all data exchanged is encrypted.

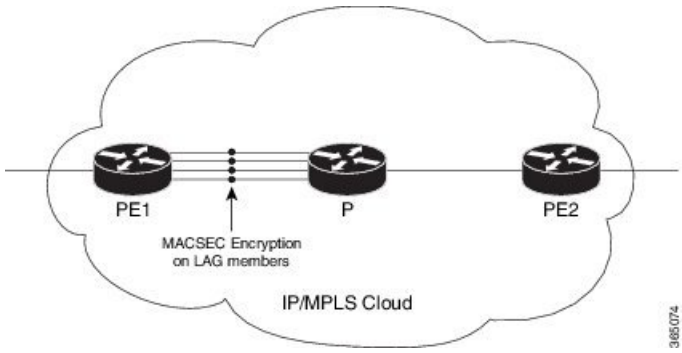
- Physical Interfaces: MACsec is configured on all router links within the MPLS core. This ensures secure data exchange across links connecting disparate data centers.

  Figure 4: MACsec on Physical Interfaces in an MPLS Core Network



- Link Bundles (LAG): When MACsec is configured on LAG members, a MACsec Key Agreement (MKA) session is established for each member. Secure Association Keys (SAK) are exchanged, allowing encryption and decryption to occur independently for each member in the group.

Figure 5: MACsec on a Link Bundle in an MPLS Core Network



MACsec encryption on Layer 3 subinterfaces

MACsec encryption on Layer 3 subinterfaces is a security mechanism that

- allows encryption and authentication of network data on VLAN-based Layer 3 subinterfaces,
- enables the application of multiple MACsec policies across different L3 subinterfaces under a single physical interface by retaining VLAN tags in clear text, and
- provides an additional security layer for communication between separate VLANs or subnets on the same physical link by making each L3 subinterface a distinct MACsec endpoint.

MACsec on Layer 3 subinterfaces uses VLAN encapsulations—802.1Q (single-tag) or 802.1ad (double-tag)—and requires specific VLAN identifiers. Keeping VLAN tags visible enables MACsec endpoints to identify subinterface traffic without encrypting the VLAN metadata. This setup allows traffic segregation at the MACsec level because each VLAN-associated subinterface has independent encryption control.

This flexibility allows for the application of different MACsec policies to Layer 3 subinterfaces under the same physical interface. By retaining unencrypted VLAN tags, Layer 3 subinterfaces can act as MACsec endpoints. Applying MACsec policies to these subinterfaces enhances network security by adding an extra layer of protection for communications between distinct subnets.

MACsec on Layer 3 subinterfaces operates similarly to that on a physical interface. For a MACsec Key Agreement (MKA) session to succeed on any Layer 3 subinterface, an appropriate tagging protocol encapsulation and a specified VLAN identifier are necessary. Although all Layer 3 subinterfaces default to 802.1Q VLAN encapsulation, the VLAN identifier must be explicitly set.

Hardware support matrix for MACsec on Layer 3 subinterfaces

Cisco IOS XR Software Release	Product ID
Release 25.3.1	8711-32FH-M
Release 25.1.1	8712-MOD-M

Cisco IOS XR Software Release	Product ID
Release 24.4.1	8608 88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM 8212-48FH-M 8711-32FH-M
Release 24.3.1	88-LC1-52Y8H-EM
Release 7.11.1	8202-32FH-M
	88-LC0-36FH-M

Guidelines for MACsec encryption on Layer 3 subinterface

Use specific encapsulation combinations

Ensure that L3 subinterfaces belonging to a physical interface utilize either 802.1Q tag (single tag) or 802.1ad outer and 802.1Q inner tags (double tags).

Consistent VLAN tagging

Configure the same type of VLAN tag on all subinterfaces associated with a physical interface.

Adhere to VLAN identifier range

MACsec encryption on a layer 3 subinterface supports a VLAN identifier range of 1–4094.

Match encapsulation and MACsec policy

The encapsulation on the L3 subinterface and the number of VLAN tags in-clear in the MACsec policy must match. If the encapsulation is 802.1Q with a single tag, the MACsec policy must reflect 1 VLAN tag in-clear. If the encapsulation is 802.1ad outer and 802.1Q inner tags, the MACsec policy must indicate 2 VLAN tags in-clear.

Configure VLAN tags in-clear

Use the **vlan-tags-in-clear** command to configure VLAN tags in-clear.

Configure encapsulation on the L3 subinterface

Use the **encapsulation dot1q** command for 802.1Q with a single tag or **encapsulation dot1ad** command for 802.1ad outer and 802.1Q inner tags.

Uniform MACsec policy parameters

All subinterfaces within a physical interface must have identical MACsec policy parameters, such as allow-lacp-in-clear, allow-pause-frames-in-clear, vlan-tags-in-clear, or security policy.

Limit MACsec sessions for optimal performance

We recommend keeping the MACsec session limit at 192 on any line card or fixed port router, including all port-level and subinterface-level MACsec sessions, to optimize simultaneous hitless SAK rekey performance.

Restrictions for MACsec encryption on Layer 3 subinterface

MACsec mutual exclusivity on physical and subinterfaces

MACsec support on physical interfaces and subinterfaces is mutually exclusive. The routers don't support simultaneously enabling MACsec on a physical interface and its subinterfaces and reject such configuration attempts. To configure MACsec on subinterfaces, clear the MACsec configurations on the corresponding physical interface and conversely.

MACsec subinterface limitation: no data delay protection

MACsec on subinterfaces does not support data delay protection.

MACsec on VLAN subinterfaces support restrictions on line cards

The Cisco 8800-LC-48H and 88-LC0-34H14FH line cards do not support MACsec on VLAN subinterfaces.

Configure MACsec encryption on VLAN subinterfaces

Enable MACsec encryption on VLAN subinterfaces using a pre-shared key chain and MACsec policies for both single-tag (802.1Q) and double-tag (802.1ad outer and 802.1Q inner) encapsulations.

Procedure

Step 1 Use Configure a MACsec keychain, on page 12 to create a MACsec key chain.

Example:

```
Router# configure
Router(config)# key chain kc
Router(config-kc)# macsec
Router(config-kc-macsec)# key 1234
Router(config-kc-macsec-1234)# key-string
12345678123456781234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm aes-256-cmac
Router(config-kc-macsec-1234)# lifetime 05:00:00 1 January 2023 infinite
Router(config-kc-macsec-1234)# commit
```

Step 2 Use Create a user-defined MACsec policy, on page 13 to create a MACsec policy.

Example:

```
802.1Q with a single tag
Router# configure
Router(config)# macsec-policy mp-SF1
Router(config-macsec-policy)# cipher-suite GCM-AES-XPN-256
Router(config-macsec-policy)# security-policy should-secure
Router(config-macsec-policy)# allow-lldp-in-clear
Router(config-macsec-policy)# key-server-priority 10
Router(config-macsec-policy)# window-size 64
Router(config-macsec-policy)# vlan-tags-in-clear 1
/* The VLAN tagging in the MACsec policy must match the encapsulation on the interface */
Router(config-macsec-policy)# commit

802.1ad outer and 802.1q inner with double tags
Router# configure
```

```
Router(config)# macsec-policy mp-SF2
Router(config-macsec-policy)# cipher-suite GCM-AES-XPN-256
Router(config-macsec-policy)# security-policy should-secure
Router(config-macsec-policy)# allow-lldp-in-clear
Router(config-macsec-policy)# key-server-priority 20
Router(config-macsec-policy)# window-size 64
Router(config-macsec-policy)# vlan-tags-in-clear 2
/* The VLAN tagging in the MACsec policy must match the encapsulation on the interface */
Router(config-macsec-policy)# commit
```

The VLAN tagging in the MACsec policy must match the encapsulation on the interface.

Step 3 Use Configure MACsec encryption on an interface, on page 15 in combination with **encapsulation dot1q** or **encapsulation dot1ad** to apply MACsec on a subinterface.

Example:

```
802.1Q with a single tag
Router# configure
Router(config)# interface HundredGigE 0/5/0/16.100
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# ipv4 address 192.168.16.1 255.255.255.0
Router(config-subif)# macsec psk-keychain kc policy mp-SF1
Router(config-subif)# commit

802.1ad outer and 802.1q inner with double tags
Router# configure
Router(config)# interface HundredGigE 0/5/0/30.200
Router(config-subif)# encapsulation dot1ad 200 dot1q 300
Router(config-subif)# ipv4 address 192.168.30.1 255.255.255.0
Router(config-subif)# macsec psk-keychain kc policy mp-SF2
Router(config-subif)# commit
```

Step 4 Use the **show running config** command to view the configurations.

MACsec key chain configurations

```
Router# show running-config psk-keychain kc
key chain kc
 macsec
  key 1234
   key-string password
11584B5643475D5B5C7B79777C6663754B56445055030F0F0B055C504C430F0F0F020006005E0D515F0905574753520C53575D72181B5F4E5D46405858517C7C7C
 cryptographic-algorithm aes-256-cmac
   lifetime 05:00:00 january 01 2023 infinite
  !
 !
!
```

MACsec policy configurations

```
802.1Q with a single tag
Router# show running-config macsec-policy mp-SF1
macsec-policy mp-SF1
 security-policy should-secure
 allow-lldp-in-clear
 window-size 64
 cipher-suite GCM-AES-XPN-256
 vlan-tags-in-clear 1
 key-server-priority 10
!
802.1ad outer and 802.1q inner with double tags
Router# show running-config macsec-policy mp-SF2
macsec-policy mp-SF2
 security-policy should-secure
```

```
 allow-lldp-in-clear
 window-size 64
 cipher-suite GCM-AES-XPN-256
 vlan-tags-in-clear 2
 key-server-priority 20
!
```

Subinterface configurations

```
802.1Q with a single tag
Router# show running-config interface HundredGigE 0/5/0/16.100
interface HundredGigE0/5/0/16.100
 ipv4 address 192.168.16.1 255.255.255.0
 macsec psk-keychain kc policy mp-SF1
 encapsulation dot1q 100
!

802.1ad outer and 802.1q inner with double tags
Router# show running-config interface HundredGigE 0/5/0/30.200
interface HundredGigE0/5/0/30.200
 ipv4 address 192.168.30.1 255.255.255.0
 macsec psk-keychain kc policy mp-SF2
 encapsulation dot1ad 200 dot1q 300
!
```

Step 5 Use **show macsec mka summary**, **show macsec policy** and **show macsec mka interface detail** commands to verify MACsec encryption on VLAN subinterfaces.

Example:

```
Router# show macsec mka summary
NODE: node0_5_CPU0
================================================================================================
   Interface-Name     Status     Cipher-Suite       KeyChain       PSK/EAP      CKN
================================================================================================
   Hu0/5/0/16.100     Secured  GCM-AES-XPN-256         kc          PRIMARY     1234
   Hu0/5/0/30.200     Secured  GCM-AES-XPN-256         kc          PRIMARY     1234
```

```
802.1Q with a single tag
Router# show macsec policy mp-SF1 detail
Policy Name                 : mp-SF1
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 10
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Should Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      LLDP In Clear         : TRUE
      Pause Frame In Clear  : FALSE
      Sak Rekey Interval    : OFF
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
      Disable Suspend On Request    : FALSE
      Disable Suspend For           : FALSE
      Enable legacy fallback        : FALSE
      SKS Profile                   : N/A
      Max AN                        : 3
      Impose Overhead on Bundle     : FALSE

802.1ad outer and 802.1q inner with double tags
Router# show macsec policy mp-SF2 detail
Policy Name                 : mp-SF2
```

```
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 20
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Should Secure
      Vlan Tags In Clear    : 2
      LACP In Clear         : FALSE
      LLDP In Clear         : TRUE
      Pause Frame In Clear  : FALSE
      Sak Rekey Interval    : OFF
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
      Disable Suspend On Request    : FALSE
      Disable Suspend For           : FALSE
      Enable legacy fallback        : FALSE
      SKS Profile                   : N/A
      Max AN                        : 3
      Impose Overhead on Bundle     : FALSE

Router# show macsec mka interface detail
Interface Name : HundredGigE0/5/0/16.100
    Interface Namestring     : HundredGigE0/5/0/16.100
    Interface short name     : Hu0/5/0/16.100
    Interface handle         : 0x2800b00
    Interface number         : 0x2800b00
    MacSecControlledIfh      : 0x2800b08
    MacSecUnControlledIfh    : 0x2800b10
    Interface MAC            : e069.bafd.e3a0
    Ethertype                : 888E
    EAPoL Destination Addr   : 0180.c200.0003
    MACsec Shutdown          : FALSE
    Config Received          : TRUE
    IM notify Complete       : TRUE
    MACsec Power Status      : Allocated
    Interface CAPS Add       : TRUE
    RxSA CAPS Add            : TRUE
    TxSA CAPS Add            : TRUE
    IM notify with VLAN Info : TRUE
    Supported VLAN encaps    : TRUE
    SecTAG Offset validation : TRUE
    VLAN                     : Outer tag (etype=0x8100, id=100, priority=0, cfi=0)
    Principal Actor          : Primary
    MKA PSK Info
      Key Chain Name         : kc
      MKA Cipher Suite       : AES-256-CMAC
      CKN                    : 12 34
    MKA fallback_PSK Info
      fallback keychain Name : - NA -
    Policy                   : mp-SF1
    SKS Profile              : N/A
    Traffic Status           : Protected
    Rx SC 1
      Rx SCI                 : e069bafde3a80064
      Rx SSCI                : 1
      Peer MAC               : e0:69:ba:fd:e3:a8
      Is XPN                 : YES
      SC State               : Provisioned
      SAK State[0]           : Provisioned
      Rx SA Program Req[0]   : 2023 Oct 27 05:41:51.701
      Rx SA Program Rsp[0]   : 2023 Oct 27 05:41:51.705
      SAK Data
        SAK[0]               : ***
```

```
        SAK Len              : 32
        SAK Version          : 1
        HashKey[0]           : ***
        HashKey Len          : 16
        Conf offset          : 0
        Cipher Suite         : GCM-AES-XPN-256
        CtxSalt[0]           : c2 b0 88 9d d6 c0 9d 3f 0a b7 99 37
        CtxSalt Len          : 12
        ssci                 : 1

    Tx SC
      Tx SCI                 : e069bafde3a00064
      Tx SSCI                : 2
      Active AN              : 0
      Old AN                 : 255
      Is XPN                 : YES
      Next PN                : 1, 0, 0, 0
      SC State               : Provisioned
      SAK State[0]           : Provisioned
      Tx SA Program Req[0]   : 2023 Oct 27 05:41:51.713
      Tx SA Program Rsp[0]   : 2023 Oct 27 05:41:51.715
      SAK Data
        SAK[0]               : ***
        SAK Len              : 32
        SAK Version          : 1
        HashKey[0]           : ***
        HashKey Len          : 16
        Conf offset          : 0
        Cipher Suite         : GCM-AES-XPN-256
        CtxSalt[0]           : c2 b0 88 9e d6 c0 9d 3f 0a b7 99 37
        CtxSalt Len          : 12
        ssci                 : 2


Interface Name : HundredGigE0/5/0/30.200
    Interface Namestring     : HundredGigE0/5/0/30.200
    Interface short name     : Hu0/5/0/30.200
    Interface handle         : 0x2800b30
    Interface number         : 0x2800b30
    MacSecControlledIfh      : 0x2800b38
    MacSecUnControlledIfh    : 0x2800b40
    Interface MAC            : e069.bafd.e410
    Ethertype                : 888E
    EAPoL Destination Addr   : 0180.c200.0003
    MACsec Shutdown          : FALSE
    Config Received          : TRUE
    IM notify Complete       : TRUE
    MACsec Power Status      : Allocated
    Interface CAPS Add       : TRUE
    RxSA CAPS Add            : TRUE
    TxSA CAPS Add            : TRUE
    IM notify with VLAN Info : TRUE
    Supported VLAN encaps    : TRUE
    SecTAG Offset validation : TRUE
    VLAN                     : Outer tag (etype=0x88a8, id=200, priority=0, cfi=0)
                             : Inner tag (etype=0x8100, id=300, priority=0, cfi=0)
    Principal Actor          : Primary
    MKA PSK Info
      Key Chain Name         : kc
      MKA Cipher Suite       : AES-256-CMAC
      CKN                    : 12 34
    MKA fallback_PSK Info
      fallback keychain Name : - NA -
    Policy                   : mp-SF2
```

```
SKS Profile              : N/A
Traffic Status           : Protected
Rx SC 1
  Rx SCI                 : e069bafde41800c8
  Rx SSCI                : 1
  Peer MAC               : e0:69:ba:fd:e4:18
  Is XPN                 : YES
  SC State               : Provisioned
  SAK State[0]           : Provisioned
  Rx SA Program Req[0]   : 2023 Oct 27 05:44:01.270
  Rx SA Program Rsp[0]   : 2023 Oct 27 05:44:01.274
  SAK Data
    SAK[0]               : ***
    SAK Len              : 32
    SAK Version          : 1
    HashKey[0]           : ***
    HashKey Len          : 16
    Conf offset          : 0
    Cipher Suite         : GCM-AES-XPN-256
    CtxSalt[0]           : 02 52 27 e4 ba 7f 16 62 52 d8 a6 e8
    CtxSalt Len          : 12
    ssci                 : 1

Tx SC
  Tx SCI                 : e069bafde41000c8
  Tx SSCI                : 2
  Active AN              : 0
  Old AN                 : 255
  Is XPN                 : YES
  Next PN                : 1, 0, 0, 0
  SC State               : Provisioned
  SAK State[0]           : Provisioned
  Tx SA Program Req[0]   : 2023 Oct 27 05:44:01.282
  Tx SA Program Rsp[0]   : 2023 Oct 27 05:44:01.284
  SAK Data
    SAK[0]               : ***
    SAK Len              : 32
    SAK Version          : 1
    HashKey[0]           : ***
    HashKey Len          : 16
    Conf offset          : 0
    Cipher Suite         : GCM-AES-XPN-256
    CtxSalt[0]           : 02 52 27 e7 ba 7f 16 62 52 d8 a6 e8
    CtxSalt Len          : 12
    ssci                 : 2
```

MACsec is enabled and secured on the specified VLAN subinterfaces. The running configuration reflects the key chain, policies, and subinterface settings, and verification outputs show the interfaces in Secured/Protected state with GCM-AES-XPN-256 and the expected policy attributes.

Alternate EAPoL Ether-type and Destination address

EAPoL Ether-types and destination addresses are WAN MACsec configuration parameters that

- identify the protocol type and destination MAC used by EAPoL frames during MACsec key agreement,
- allow alternate values to prevent Layer 2 intermediate devices from consuming EAPoL packets, and

- support per-interface and per-subinterface configuration with inheritance from the parent interface to improve reliability and flexibility.
- EAPoL: Extensible Authentication Protocol over LAN; the protocol that transports MACsec Key Agreement (MKA) control traffic at Layer 2.
- Ether-type: A 16-bit field in an Ethernet frame that indicates the upper-layer protocol carried (for EAPoL, the standard value is 0x888E).
- Destination MAC address: The Layer 2 address used to deliver EAPoL frames (for EAPoL, the standard multicast address is 01:80:C2:00:00:03).

In WAN MACsec deployments, utilizing the standard EAPoL Ether-Type (0x888E) and destination MAC address (01:80:C2:00:00:03) can result in intermediate Layer 2 devices intercepting and consuming EAPoL packets across a service provider network. To prevent such interference and enhance MACsec session establishment between peers, configuration of an alternate EAPoL Ether-Type, an alternate destination MAC address, or both, on a MACsec-enabled interface, is recommended.

- Alternate EAPoL Ether-type: The supported alternate Ether-type is 0x876F. This can be configured to avoid packet interception.
- Alternate destination MAC address: Options include using the broadcast address FF:FF:FF:FF:FF or the nearest bridge group address. This configuration helps in reducing interference.
- Subinterface configuration: Specific EAPoL parameters can be explicitly set for each subinterface. If not set, subinterfaces will inherit the EAPoL configuration from the parent physical interface.

This structured approach ensures a reliable and interference-free MACsec deployment across WAN environments.

Table 5: Hardware Support Matrix for alternate EAPoL Ether-type and Destination address

Cisco IOS XR Software Release	Product ID
Release 25.4.1	8711-32FH-M
Release 25.3.1	88-LC1-52Y8H-EM 8212-48FH-M
Release 7.10.1	8608
Release 7.5.2	8202-32FH-M
Release 7.3.3	8-LC0-34H14FH
Release 7.3.15	88-LC0-36FH-M
Release 7.0.12	88-LC-48H

Configure EAPoL Ether-type 0x876F

Configure the EAPoL Ether-type 0x876F on a router interface to enable enhanced authentication protocols.

This task involves setting up the EAPoL Ether-type and applying MACsec on an interface to ensure secure communication.

Procedure

Step 1 Use Configure a MACsec keychain, on page 12 to create a MACsec key chain.

Step 2 (Optional) Use Create a user-defined MACsec policy, on page 13 to create a MACsec policy.

Step 3 Use **eapol eth-type 876F** to configure the EAPoL ether-type.

Example:

```
Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# eapol eth-type 876F
Router(config-if)# commit
```

Step 4 Use Configure MACsec encryption on an interface, on page 15 command to apply MACsec on a interface.

Example:

```
Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# macsec psk-keychain kc fallback-psk-keychain fb
Router(config-if)# commit
```

Step 5 Use the **show running config** command to view the configurations.

Example:

```
Router# show running-config interface HundredGigE0/1/0/2
interface HundredGigE0/1/0/2
  eapol eth-type 876F
  macsec psk-keychain kc fallback-psk-keychain fb
!
```

Step 6 Use **show macsec mka summary** and **show macsec mka session** commands to verify EAPoL Ether-type 0x876F on an interface.

Example:

```
Router# show macsec mka interface  HundredGigE0/1/0/2 detail | i Ethertype
Ethertype                : 876F

Router# show macsec mka session interface HundredGigE0/1/0/2.1
================================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
================================================================================================
     Hu0/1/0/2        0201.9ab0.77cd/0001     1      Secured      YES       PRIMARY     1234
     Hu0/1/0/2        0201.9ab0.77cd/0001     1      Active       YES     FALLBACK      9999
```

The EAPoL Ether-type 0x876F is configured and MACsec is applied to the specified interface.

Configure EAPoL destination broadcast address

Configure the EAPoL destination address to use the broadcast address FF:FF:FF:FF:FF to ensure EAPoL packets are flooded to all receivers in the underlying L2 network

This task involves setting the EAPoL destination address to broadcast and applying MACsec on an interface for secure communication.

Procedure

Step 1 Use Configure a MACsec keychain, on page 12 to create a MACsec key chain.

Step 2 (Optional) Use Create a user-defined MACsec policy, on page 13 to create a MACsec policy.

Step 3 Use **eapol destination-address broadcast-address** command to configure the EAPoL destination address to broadcast.

Example:

```
Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# eapol destination-address broadcast-address
Router(config-if)# commit
```

Step 4 Use Configure MACsec encryption on an interface, on page 15 to apply MACsec on a interface.

Example:

```
Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# macsec psk-keychain kc fallback-psk-keychain fb
Router(config-if)# commit
```

Step 5 Use the **show running config** command to view the EAPoL destination address to broadcast configurations.

Example:

```
Router# show running-config interface HundredGigE0/1/0/2
interface HundredGigE0/1/0/2
   eapol destination-address ffff.ffff.ffff
 macsec psk-keychain kc fallback-psk-keychain fb
!
```

Step 6 Use **show macsec mka summary** and **show macsec mka session** commands to verify EAPoL destination address to broadcast on an interface.

Example:

```
Router# show macsec mka interface  HundredGigE0/1/0/2 detail  | i EAPoL
    EAPoL Destination Addr   : ffff.ffff.ffff

Router# show macsec mka session interface HundredGigE0/1/0/2
================================================================================================
   Interface-Name        Local-TxSCI        #Peers   Status   Key-Server   PSK/EAP      CKN
================================================================================================
     Hu0/1/0/2       02df.3638.d568/0001     1      Secured      YES      PRIMARY     1234
     Hu0/1/0/2       02df.3638.d568/0001     1      Active       YES     FALLBACK     9999
```

The EAPoL destination address is configured to broadcast, and MACsec is applied to the specified interface.

Configure EAPoL destination bridge group address

Set the EAPoL destination address to the nearest bridge group address (e.g., 01:80:C2:00:00:00) on a physical interface, with the configuration inherited by the MACsec-enabled subinterface.

This task involves configuring the EAPoL destination address on a physical interface and applying MACsec to a subinterface for enhanced security.

Procedure

Step 1 Use Configure a MACsec keychain, on page 12 to create a MACsec key chain.

Step 2 (Optional) Use Create a user-defined MACsec policy, on page 13 to create a MACsec policy.

Step 3 Use **eapol destination-address bridge-group-address** command to configure the EAPoL destination bridge group address on a MACsec-enabled physical interface.

Example:

```
Router(config)# interface HundredGigE0/1/0/1
Router(config-if)# eapol destination-address bridge-group-address
Router(config-if)# commit
```

Step 4 Use Configure MACsec encryption on an interface, on page 15 to apply MACsec on a interface.

Example:

```
Router(config)# interface HundredGigE0/1/0/1.1
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# macsec psk-keychain kc fallback-psk-keychain fb
outer(config-subif)# commit
```

Step 5 Use the **show running config** command to view the configurations.

Example:

This example shows the running configuration for the EAPoL destination bridge group address on the MACsec-enabled physical interface.

```
Router# show running-config interface Hu0/1/0/1
interface HundredGigE0/1/0/1
eapol destination-address 0180.c200.0000
```

This example shows the running configuration for the EAPoL destination bridge group address on the MACsec-enabled subinterface.

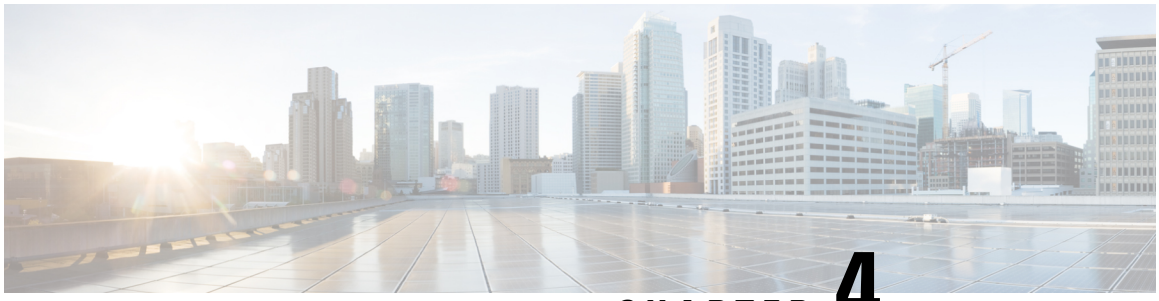
```
Router# show running-config interface HundredGigE0/1/0/1.1
interface HundredGigE0/1/0/0.1
  macsec psk-keychain kc fallback-psk-keychain fb
  encapsulation dot1q 1
!
```

Step 6 Use **show macsec mka summary** and **show macsec mka session** commands to verify APoL destination bridge group address on the MACsec-enabled subinterface.

Example:

```
Router# show macsec mka interface  HundredGigE0/1/0/1.1 detail  | i EAPoL
    EAPoL Destination Addr   : 0180.c200.0000

Router# show macsec mka session interface HundredGigE0/1/0/1.1
==============================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
==============================================================================================
     Hu0/1/0/1.1       0201.9ab0.85af/0001     1      Secured      YES      PRIMARY     1234
     Hu0/1/0/1.1       0201.9ab0.85af/0001     1      Active       YES     FALLBACK     9999
```

The EAPoL destination bridge group address is configured, and MACsec is applied to the specified subinterface.



CHAPTER 4

MACsec policy exceptions

This chapter explains how to configure MACsec policy exceptions to permit specific packet types—such as LACP, pause frames, and LLDP packets—to bypass MACsec encryption and be transmitted in clear text. It provides step-by-step procedures, example commands, and important security considerations for enabling these exceptions in Cisco environments.

MACsec policy exception

A MACsec policy exception is a mechanism within a MACsec security policy that

- bypasses MACsec encryption or decryption for specific data packets,
- allows these packets to be sent in clear-text format, and
- supports interoperability scenarios and certain network topologies.

By default, a MACsec security policy uses the **must-secure** option, which mandates data encryption for all traffic. Specific commands can optionally bypass MACsec encryption or decryption, enabling certain packet types to be transmitted in clear text.

Within the **macsec-policy** configuration mode, the **allow** option is available to permit clear-text transmission for designated packet types.

Table 6: MACsec Policy Options: Must-Secure vs. Allow

Feature / Behavior	must-secure	allow
Encryption enforcement	Required for all traffic	Mandatory except for packets explicitly allowed
Use case	Provides maximum security	Allows interoperability in mixed environments
Packet exceptions	Not permitted	Specific packet types can bypass encryption

Feature / Behavior	must-secure	allow
Example commands	N/A	`allow lacp-in-clear` `allow pause-frames-in-clear`
Security level	Highest (no clear-text transmission)	Slightly reduced (clear text allowed for selected frames)

MACsec policy exceptions

- Using the `allow lacp-in-clear` command to bypass MACsec for Link Aggregation Control Protocol (LACP) packets. This is beneficial in scenarios where bundles are terminated at an intermediate node and MACsec is enforced only at end nodes or when the remote node expects clear text.
- Using the `allow pause-frames-in-clear` command to transmit Ethernet PAUSE frame packets in clear text.

Create a MACsec policy exception

Allow specific MACsec policy exceptions to enable or permit particular packet types in clear-text format.

Procedure

Step 1 Use the **macsec-policy** command to access the desired MACsec policy configuration by specifying the policy name.

Example:

```
Router# configure
Router(config)# macsec-policy mp1
```

Step 2 Use the **allow lacp-in-clear** command to permit LACP packets in clear-text format.

Example:

```
Router(config-macsec-policy)# allow lacp-in-clear
```

Step 3 Use the **allow pause-frames-in-clear** command to permit pause frames in clear-text.

Example:

```
Router(config-macsec-policy)# allow pause-frames-in-clear
Router(config-macsec-policy)# commit
```

Step 4 Use the **show running config** command to confirm the policy exception.

Example:

```
Router# show running-config macsec-policy mp1
macsec-policy mp1
…
allow lacp-in-clear
allow pause-frames-in-clear
!
```

Step 5 Use the **show macsec policy detail** command to verify detailed MACsec policy status.

Example:

```
Router# show macsec policy detail
Total Number of Policies = 1
------------------------------------------------------
Policy Name : mp1
Cipher Suite : GCM-AES-XPN-256
Key-Server Priority : 10
Window Size : 64
Conf Offset : 50
Replay Protection : TRUE
Delay Protection : FALSE
Security Policy : Must Secure
Vlan Tags In Clear : 1
LACP In Clear : TRUE
LLDP In Clear : FALSE
Pause Frame In Clear : TRUE
Sak Rekey Interval : 60 seconds
```

The MACsec policy is updated to allow the specified packet exceptions in clear text, using the recommended **allow** commands for new configurations.

MACsec policy exceptions for LACP packets

A MACsec policy exception for LACP packets is a network security configuration that

- permits LACP packets to bypass MACsec encryption and be sent in clear text,
- is used in scenarios where LACP bundles terminate at intermediate network nodes while MACsec is only enforced at end nodes, and
- supports interoperability where remote nodes expect LACP packets in clear text.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
MACsec Policy Exception forLink Aggregation Control Protocol Packets	Release 26.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*); *This feature is supported on: • 8712-MOD-M router. • 8011-4G24Y4H-I • 8011-32Y8L2H2FH • 8011-12G12X4Y-D/A

Feature Name	Release Information	Feature Description
MACsec Policy Exception for Link Aggregation Control Protocol Packets	Release 7.0.12	We have introduced an option in MACsec policy exceptions to accommodate Link Aggregation Control Protocol (LACP) packets in an unencrypted format. LACP packets sent in clear text enable seamless bundle formation and troubleshooting of link aggregation issues on MACsec-enabled ports. By default, MACsec operates in must-secure mode, permitting encrypted traffic flow and LACP packets only after securing the MACsec Key Agreement (MKA) session. The LACP-in-clear feature allows LACP packets to bypass MACsec encryption, ensuring compatibility with intermediate nodes and supporting interoperability scenarios where the remote device expects LACP packets in clear text. CLI: • The **lacp-in-clear** keyword is introduced in the **allow** command.

MACsec policy exceptions for LLDP packets

A MACsec policy exception for LLDP packets is a MACsec configuration mechanism that

- allows the transmission of LLDP (Link Layer Discovery Protocol) packets in clear text even when MACsec encryption is enabled,
- enables network administrators to facilitate neighbor discovery and troubleshooting by making LLDP packets visible on the network, and
- maintains MACsec encryption for all other traffic, ensuring the overall security of the data link layer.

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
MACsec Policy Exception for Link Layer Discovery Protocol Packets	Release 26.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*); *This feature is supported on: • 8712-MOD-M router. • 8011-4G24Y4H-I • 8011-32Y8L2H2FH • 8011-12G12X4Y-D/A
MACsec Policy Exception for Link Layer Discovery Protocol Packets	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M

Feature Name	Release Information	Feature Description
MACsec Policy Exception for Link Layer Discovery Protocol Packets	Release 7.11.1	We have introduced an option in MACsec policy exceptions to accommodate Link Layer Discovery Protocol (LLDP) packets in an unencrypted format. LLDP packets in clear text format help you troubleshoot LLDP neighbor discovery network issues on MAcsec-enabled ports. By default, MACsec always operates in must-secure mode, allowing encrypted traffic flow including LLDP packets only after securing the MACsec Key Agreement (MKA) session. The feature introduces these changes: CLI: • The **lldp-in-clear** keyword is introduced in the **allow** command. • The **lldp-in-clear** status is displayed in the **show macsec policy detail** command. • The **lldp-in-clear** status is displayed in the **show macsec mka interface detail** command. YANG Data Models: • `Cisco-IOS-XR-crypto-macsec-mka-cfg.yang` • `Cisco-IOS-XR-crypto-macsec-mka-oper.yang` • `Cisco-IOS-XR-um-macsec-cfg.yang` (See GitHub, YANG Data Models Navigator)

By default, when MACsec is enabled, it encrypts all network traffic at the data link layer, including LLDP packets. This ensures that communications between peers remain secure. The router stores information learned from LLDP exchanges in the Management Information Base (MIB).

However, starting from Cisco IOS XR Software Release 7.11.1, routers provide an option to transmit LLDP packets in clear text even when MACsec is enabled. Administrators can enable this exception using the **allow lldp-in-clear** command in the MACsec policy. This functionality is useful for troubleshooting LLDP neighbor

discovery issues, as it allows network administrators to view LLDP packets unencrypted for diagnostic purposes.

By default, MACsec operates in must-secure mode, permitting encrypted traffic flow—including LLDP packets—only after the MACsec Key Agreement (MKA) session is secured.

Note We strongly advise against enabling the MACsec exception to retain LLDP packets unencrypted unless necessary for network maintenance. You must ensure to configure LLDP packets in clear text at both ends of the MACsec link.

Table 9: LLDP packet handling in MACsec: default encryption versus clear-text exception

Feature	Default MACsec behavior	With LLDP exception enabled
LLDP packet encryption	Encrypted	Clear text (unencrypted)
Troubleshooting support	Limited (due to encryption)	Enhanced (LLDP packets visible in plain text)
Security posture	Highest (all packets encrypted)	Slightly reduced for LLDP, others encrypted

LLDP packet handling in MACsec: default encryption and clear-text exception

- An administrator enables the **allow lldp-in-clear** command to transmit LLDP packets in clear text on a MACsec-enabled port for troubleshooting neighbor discovery issues.
- By default, a router encrypts LLDP packets with MACsec, ensuring all data link layer communications are secured.

MACsec LLDP clear-text: troubleshooting challenges and security risks

- Without enabling the LLDP exception, LLDP packets remain encrypted under MACsec, making it difficult to diagnose neighbor discovery problems using packet captures.
- Transmitting all protocol packets, including LLDP, in clear text on a MACsec-enabled port would compromise the security provided by MACsec, which is not the default or recommended configuration.

Configure MACsec policy exception for LLDP packets

Configure a MACsec policy to allow LLDP packets to be transmitted in clear-text format without encryption.

By default, MACsec encrypts all traffic on a link. This task enables an exception for Link Layer Discovery Protocol (LLDP) packets, allowing them to pass through unencrypted while maintaining encryption for other traffic types.

Procedure

Step 1 Use the **macsec-policy** command to access the desired MACsec policy configuration by specifying the policy name.

Example:

```
Router# configure
Router(config)# macsec-policy test-macsec
```

Step 2 Use the **allow lldp-in-clear** command to enable the LLDP clear-text exception.

Example:

```
Router(config-macsec-policy)# allow lldp-in-clear
```

Step 3 Use the **show running config** command to confirm the policy exception.

Example:

```
Router# show running-config macsec-policy test-macsec
macsec-policy mp1
…
allow lldp-in-clear
!
```

Step 4 Use the **show macsec policy detail** and **show macsec mka interface detail** commands to verify the policy details reflect the LLDP clear-text setting.

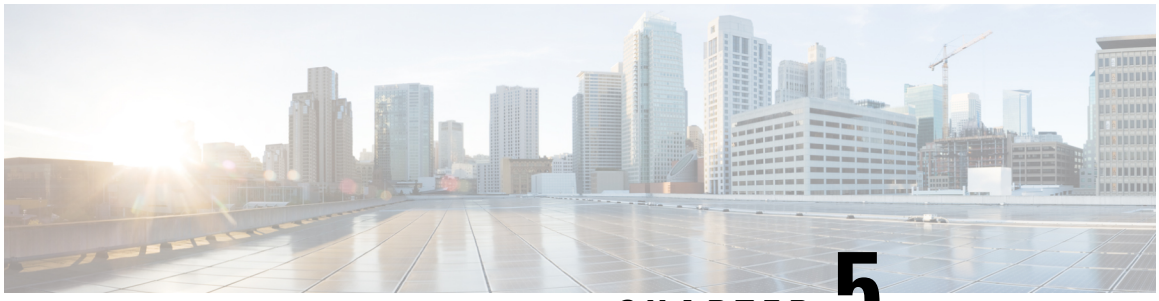
Example:

```
Router# show macsec policy detail
Total Number of Policies = 1
------------------------------------------------------
Policy Name : mp1
Cipher Suite : GCM-AES-XPN-256
Key-Server Priority : 10
Window Size : 64
Conf Offset : 50
Replay Protection : TRUE
Delay Protection : FALSE
Security Policy : Must Secure
Vlan Tags In Clear : 1
LACP In Clear : FALSE
LLDP In Clear : TRUE
Pause Frame In Clear : FALSE
Sak Rekey Interval : 60 seconds
```

```
Router# show macsec mka interface detail
Number of interfaces on node node0_3_CPU0 : 1
------------------------------------------------------
Interface Name : HundredGigE0/3/0/5
Interface Namestring : HundredGigE0/3/0/5
Interface short name : Hu0/3/0/5
Interface handle : 0x1800238
Interface number : 0x1800238
MacSecControlledIfh : 0x18005e0
MacSecUnControlledIfh : 0x18005e8
Interface MAC : 5cb1.2ede.7648
Ethertype : 888E
EAPoL Destination Addr : 0180.c200.0003
MACsec Shutdown : FALSE
Config Received : TRUE
IM notify Complete : TRUE
```

```
MACsec Power Status : Allocated
Interface CAPS Add : TRUE
RxSA CAPS Add : TRUE
TxSA CAPS Add : TRUE
lldp-in-clear : TRUE
```

The MACsec policy is updated to allow LLDP packets in clear-text, confirmed by showing running configuration and interface details.



CHAPTER 5

MACsec encryption using EAP-TLS authentication

This chapter provides step-by-step guidance on configuring MACsec encryption using EAP-TLS authentication on the routers. It covers how the process works, key roles and components involved, best practice guidelines, configuration procedures, and verification commands to ensure secure, certificate-based Ethernet traffic encryption.

MACsec encryption using EAP-TLS authentication

MACsec encryption using EAP-TLS authentication is a Ethernet traffic securing method that

- provides Media Access Control Security (MACsec) encryption between two routers using IEEE 802.1X port-based authentication,
- enables mutual authentication between the authentication server and client with Extensible Authentication Protocol–Transport Layer Security (EAP-TLS) certificates, and
- derives the Master Session Key (MSK), Connectivity Association Key (CAK), and Connectivity Association Key Name (CKN) from the EAP-TLS authentication process for establishing MACsec encryption.

IEEE 802.1X device roles

The devices in the network play specific roles during IEEE 802.1X authentication.

- Supplicant: An entity at one end of a point-to-point LAN segment that seeks authentication by an Authenticator attached to the other end of that link.
- Authenticator: An entity that facilitates authentication of other entities attached to the same LAN.

- Authentication server: An entity that provides an authentication service to an Authenticator. The service determines whether the Supplicant is authorized to access system services where the Authenticator resides by evaluating the credentials provided by the Supplicant.

How MACsec encryption using EAP-TLS authentication works

MACsec encryption using EAP-TLS authentication establishes secure communication between routers by leveraging certificate-based mutual authentication to derive keys for MACsec encryption.

Summary

The key components involved in the process are:

- Routers (authenticator/supplicant): Systems that perform MACsec encryption and participate in 802.1X authentication, acting as either the authenticator (facilitates authentication) or the supplicant (seeks authentication).
- Authentication server (RADIUS/Cisco ISE/ACS): An entity that provides authentication services to an authenticator, verifying supplicant credentials and facilitating EAP-TLS communication.
- Certificate Authority (CA) server: Issues and manages digital certificates used for mutual authentication in EAP-TLS.
- EAP-TLS (Extensible Authentication Protocol-Transport Layer Security): The authentication method used for mutual authentication between the authentication server and the client (supplicant) using certificates.
- Master Session Key (MSK): A cryptographic key generated upon successful EAP-TLS authentication.
- Connectivity Association Key (CAK): Derived from the MSK, this key is used by the MACsec Key Agreement (MKA) protocol.
- Connectivity Association Key Name (CKN): Derived from the EAP session ID, this name identifies the CAK.

Workflow

These stages describe how MACsec encryption using EAP-TLS authentication works:

1. Initiation: A supplicant router initiates 802.1X port-based authentication on a physical Ethernet interface with an authenticator router.
2. EAP message exchange: The authenticator router forwards EAP messages between the supplicant and the configured external authentication server (e.g., RADIUS) using EAP as the transport.
3. Mutual authentication (EAP-TLS): The authentication server and the supplicant router perform mutual authentication using digital certificates via the EAP-TLS method. This requires both devices to have valid certificates issued by a trusted Certificate Authority.
4. Master session key generation: Upon successful EAP-TLS authentication, a Master Session Key (MSK) is generated.
5. Key derivation: The MSK is then used to derive the Connectivity Association Key (CAK), and the Connectivity Association Key Name (CKN) is derived from the EAP session ID.

6. MACsec Key Agreement (MKA): The derived CAK and CKN are utilized by the MKA protocol to establish and maintain secure MACsec encryption between the routers on the interface.

Result

This process enables robust MACsec encryption between two routers, ensuring data confidentiality and integrity on Ethernet interfaces through secure, certificate-based authentication and automated key management.

Guidelines for MACsec encryption using EAP-TLS authentication

- Ensure that you use 802.1X only on physical Ethernet interfaces when configuring EAP-TLS authentication.
- Use 802.1X port-based authentication exclusively to derive keys for MACsec Key Agreement (MKA). The authentication process does not perform port control functions.
- Configure the router in the Authenticator or Supplicant Port Access Entity (PAE) role. The router supports both roles.
- As an authenticator, ensure that remote EAP authentication uses RADIUS as the EAP transport.
- The router supports EAP-TLS authentication in single-host mode only, as it does not support multi-host mode.

Configure MACsec encryption using EAP-TLS authentication

Securely authenticate 802.1X clients and enable MACsec encryption on the router using EAP-TLS.

This task enables the router to authenticate 802.1X clients with EAP-TLS, providing mutual authentication and generating a Master Session Key (MSK) for secure communication.

Before you begin

- Ensure a Certificate Authority (CA) server is configured for the network.
- Verify the configured CA certificate is valid.
- Confirm that Cisco Identity Services Engine (ISE) Release 2.2 or later, or Cisco Secure Access Control Server Release 5.6 or later, is configured as the external AAA server.
- Ensure the remote AAA server is configured with the EAP-TLS method.
- Synchronize the routers, CA server, and external AAA server using Network Time Protocol (NTP) to ensure certificate validation.

Follow these steps to configure MACsec encryption using EAP-TLS authentication:

Procedure

Step 1 Configure the RADIUS server pre-shared keys.

Example:

```
Router# config
Router(config)# radius-server host 209.165.200.225 key 7 094F471A1A0A57
Router(config)# radius-server vsa attribute ignore unknown
Router(config)# commit
```

Step 2 Configure the 802.1X authentication method using RADIUS as the protocol.

Example:

```
Router# config
Router(config)# aaa authentication dot1x default group radius
Router(config)# commit
```

Step 3 Generate an RSA key pair to sign and encrypt key management messages.

Example:

```
Router# config
Router(config)# crypto key generate rsa 8002
Wed Aug 7 10:25:22.461 UTC
The name for the keys will be: 8002
Choose the size of the key modulus in the range of 512 to 4096 for your General Purpose Keypair.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [2048]: 600
Generating RSA keys ... Done w/ crypto generate keypair
[OK]
```

Step 4 Configure a trustpoint to manage and track CAs and certificates.

Example:

```
Router# config
Router(config)# crypto ca trustpoint test2
Router(config-trustp)# enrollment url http://caurl.com
Router(config-trustp)# subject-name CN=8000Series,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Router(config-trustp)# rsakeypair 8002
Router(config-trustp)# crl optional
Router(config-trustp)# commit
```

Step 5 Configure a domain name for certificate enrollment.

Example:

```
Router# config
Router(config)# domain name ca.8000-series.cisco.com
Router(config)# commit
```

Step 6 Authenticate the CA and enroll the device certificate.

Example:

```
Router# config
Router(config)# crypto ca authenticate test2
Router(config)# crypto ca enroll test2
Router(config)# commit
```

Step 7 Configure an EAP profile.

Example:

```
Router# config
Router(config)# eap profile 8002
Router(config-eap)# identity CE1
Router(config-eap)# method tls pki-trustpoint test2
Router(config-eap)# commit
```

Step 8 Configure an 802.1X profile on the device.

Example:

```
Router# config
Router(config)# dot1x profile 8k_prof
Router(config-dot1x-8k_prof)# pae both
Router(config-dot1x-8k_prof)# authenticator timer reauth-time 3600
Router(config-dot1x-8k_prof)# supplicant eap profile 8002
Router(config-dot1x-8k_prof)# exit
Router(config)# commit
```

Step 9 Apply the MACsec EAP profile and the 802.1X profile to an interface.

Example:

```
Router# config
Router(config)# interface fourHundredGigE 0/0/0/0
Router(config-if)# dot1x profile 8k_prof
Router(config-if)# macsec eap policy macsec-1
Router(config-if)# commit
```

MACsec encryption is successfully configured on the router using EAP-TLS authentication, enabling secure communication and mutual authentication for 802.1X clients.

Verify MACsec encryption and 802.1X configuration on an interface

Validate the status and configuration details of MACsec EAP and 802.1X on a router interface.

Perform validation during security audits, after deployment, or after making configuration changes.

Procedure

Step 1 Use the **show dot1x interface detail** command to view detailed 802.1X information for the interface.

Example:

```
Router# show dot1x interface HundredGigE 0/0/0/24 detail
Dot1x info for HundredGigE 0/0/0/24
------------------------------------------------------------
Interface short name        : Hu0/0/0/24
Interface handle            : 0x800020
Interface MAC               : 0201.9ab0.85af
Ethertype                   : 888E
PAE                         : Both
Dot1x Port Status           : AUTHORIZED
Dot1x Profile               : 8k_prof
Supplicant:
 Config Dependency          : Resolved
 Eap profile                : 8k
 Client List:               : 0257.3fae.5cda
 Authenticator EAP Method   : EAP-TLS
 Supp SM State              : Authenticated
 Supp Bend SM State         : Idle
 Last authen time           : 2018 Mar 01 13:31:03.380
Authenticator:
 Config Dependency          : Resolved
 ReAuth                     : Enabled, 0 day(s), 01:00:00
```

```
 Client List:              : 0257.3fae.5cda
 Auth SM State             : Authenticated
 Auth Bend SM State        : Idle
 Last authen time          : 2018 Mar 01 13:33:17.852
 Time to next reauth       : 0 day(s), 00:59:57
MKA Interface:
 Dot1x Tie Break Role      : Auth
 EAP Based Macsec          : Enabled
 MKA Start time            : 2018 Mar 01 13:33:17.852
 MKA Stop time             : NA
 MKA Response time         : 2018 Mar 01 13:33:18.357
```

In the**show dot1x interface detail** command output, check for these status indicators.

- Confirm that the **Dot1x Port Status** is **AUTHORIZED**.
- Verify the EAP method and the authentication state of the client.
- Check the last authentication time and related status indicators.

Step 2 Use the **show macsec mka session interface** command to view MACsec MKA session status.

Example:

```
Router# show macsec mka session interface HundredGigE 0/0/0/24

=======================================================================
Interface    Local-TxSCI           # Peers  Status    Key-Server
=======================================================================
Hu0/0/0/24   0201.9ab0.85af/0001   1        Secured   YES
```

Ensure the **Status** is **Secured** and that **Key-Server** is **YES**.

Step 3 Use the **show macsec mka session interface detail** command to view detailed MACsec MKA session information.

Example:

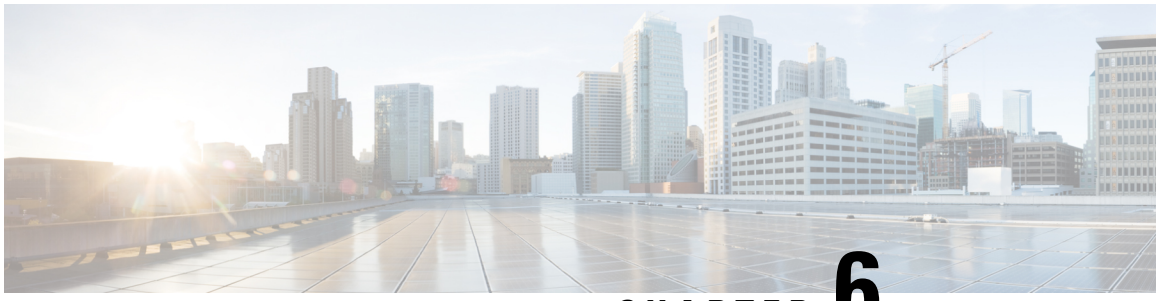
```
Router# show macsec mka session interface HundredGigE 0/0/0/24 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI                 : 0201.9ab0.85af/0001
Local Tx-SSCI                : 2
Interface MAC Address        : 0201.9ab0.85af
MKA Port Identifier          : 1
Interface Name               : Hu0/0/0/24
CAK Name (CKN)               : A94399EE68B2A455F85527A4309485DA
CA Authentication Mode       : EAP
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
 Last Rx MKPDU               : 2018 Mar 01 13:36:56.450
 Peer Count                  : 1
 RxSCI                       : 02573FAE5CDA0001
 MI                          : 86B47DE76B42D9D7AB6805F7
 Peer CAK                    : Match
```

In the **show macsec mka session interface detail** command output, verify these session aspects.

- Verify the session status is SECURED.
- Check the local SCI (Secure Channel Identifier) value and the peer SCI value.
- Confirm the cipher suite used (e.g., AES-128-CMAC, GCM-AES-XPN-256).
- Review the live peer list and the MKA policy details.

You will have validated that MACsec and 802.1X are properly configured and operational on the specified interface.



CHAPTER 6

MACsec encryption using SKIP

This chapter provides guidance on configuring point-to-point MACsec encryption using the Secure Key Integration Protocol (SKIP) with Quantum Key Distribution (QKD) devices. It covers protocol overview, configuration steps, supported topologies, and key operational considerations for achieving quantum-safe key management on the routers.

Secure Key Integration Protocol

A Secure Key Integration Protocol is a protocol that

- enables routers to communicate with external quantum devices
- facilitates the exchange of MACsec encryption keys using Quantum Key Distribution (QKD), and
- addresses the key distribution problem in a post-quantum world.

A Quantum Key Distribution (QKD) is a cryptographic technique that

- uses the laws of quantum mechanics to ensure secure transmission of a secret key between two parties
- encodes the key in the quantum states of single photons and transmits it over optical fiber or free space (vacuum), and
- provides security by making any interception detectable, since measuring a quantum state changes it, thus alerting the communicating parties to eavesdropping attempts.

QKD is resistant to quantum attacks and is expected to remain secure even as cryptanalysis and quantum computing advance. Unlike traditional cryptographic algorithms, QKD does not require continual updates in response to new vulnerabilities.

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Secure Key Integration Protocol for Routers	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on the Cisco 8712-MOD-M routers.
Secure Key Integration Protocol for Routers	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M
Secure Key Integration Protocol for Routers	Release 7.9.1	Your routers are now capable of handling the Secure Key Integration Protocol (SKIP) protocol. The SKIP protocol enables your routers to communicate with external quantum devices. With this ability, you can use the Quantum Key Distribution (QKD) devices for exchanging MACsec encryption keys between routers. Using QKD eliminates the key distribution problem in a post quantum world where the current cryptographic systems are no longer secure due to the advent of quantum computers. This feature introduces the following: • CLI: • **crypto-sks-kme** • **show crypto sks profile** • Yang Data Model: Cisco-IOS-XR-um-sks-server-cfg.yang (see GitHub, YANG Data Models Navigator) For more information on Quantum Key Distribution, see Post Quantum Security Brief.

Supported configuration strategies for QKD devices

Secure Key Integration Protocol allows various configurations for utilizing QKD devices:

- Single QKD device configuration: Use the same QKD device at the end ports of the peer routers to exchange encryption keys efficiently.

- Multiple QKD device configuration: Configure different QKD devices on the end ports of peer routers for improved flexibility and security.
- Multi-link QKD device detup: Establish multiple communication links between the same peer routers using different QKD devices for enhanced security.

Options for router communication with QKD devices

To ensure efficient and secure integration between routers and Quantum Key Distribution (QKD) devices, certain router configurations are recommended.

These options optimize routing communication with QKD devices:

- Source interface configuration: Specify an explicit source interface for QKD device communication using the source interface command within the SKS (Secure Key Service) profile settings. Defining the source interface controls which interface initiates outbound communication and is critical for both security and routing policies.

```
Router# config
Router(config)# sks profile ProfileR1toR2 type remote
Router(config-sks-profile)# kme server ipv4 192.0.2.34 port 10001
Router(config-sks-profile)# source interface hundredGigE 0/1/0/17
Router(config-sks-profile)# commit
```

- HTTP proxy configuration: In environments requiring proxy intermediaries, configure routers to use an HTTP proxy when communicating with QKD devices. The http proxy server command allows specifying the IPv4 or IPv6 proxy address or hostname and the required TCP port.

```
Router# config
Router(config)# sks profile ProfileR1toR2 type remote
Router(config-sks-profile)# kme server ipv4 192.0.2.34 port 10001
Router(config-sks-profile)# http proxy ipv4 192.0.2.68 port 804
Router(config-sks-profile)# commit
```

How point-to-point MACsec encryption using SKIP works

Point-to-point MACsec encryption establishes secure communication between peer router interfaces by leveraging an external quantum key distribution (QKD) network for key exchange. This approach ensures secure and automated key management.

Summary

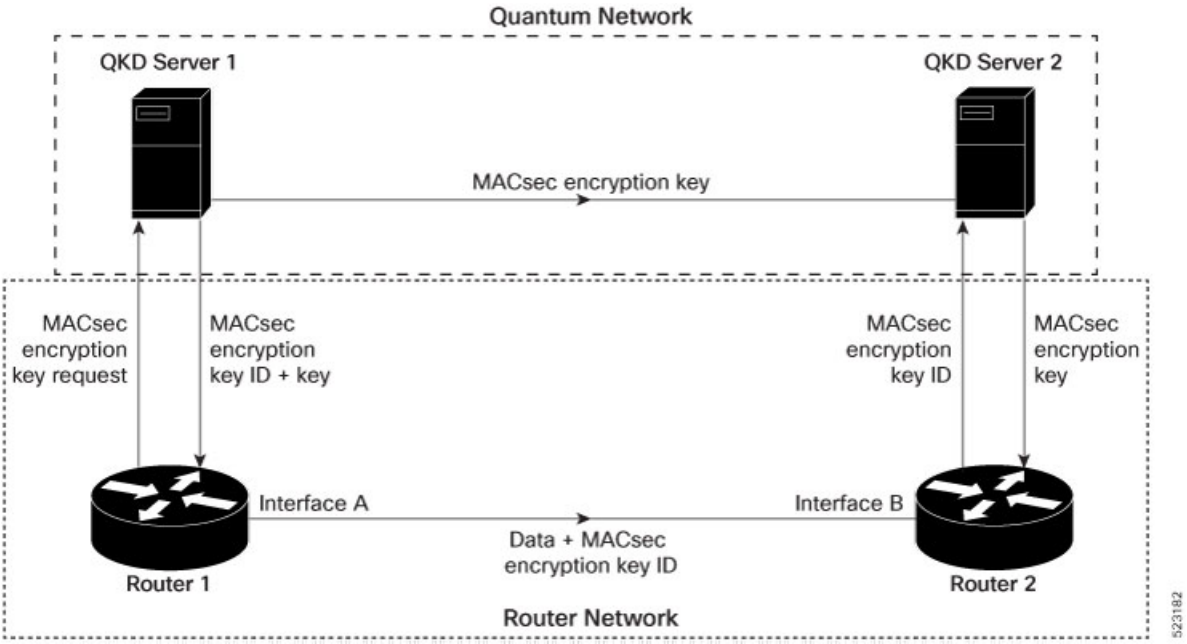
The key components involved in the process are:

- Router: Initiates the MACsec link creation and communicates with the QKD device using SKIP.
- Peer router: The other end of the MACsec link, which also communicates with its QKD device.
- SKIP: The protocol a router uses to establish secure encryption.
- External QKD device network: A network of Quantum Key Distribution devices responsible for securely sharing MACsec encryption keys.

- QKD device: A specific device within the QKD network that generates key pairs (key ID and key) and shares them.
- Key ID: A unique string that identifies the shared secret (key).
- Key (shared secret): The actual MACsec encryption key.

Workflow

Figure 6: Point-to-point MACsec Link Encryption using SKIP



The process involves the following stages:

1. Link creation request: A router needs to create a MACsec link between its interface and a peer router's interface.
2. Key request to QKD: The router contacts its external QKD device and requests the encryption key.
3. Key pair generation: The external QKD device generates a key pair. This pair comprises a unique key ID and the encryption key.
4. Key distribution to initiating router: The QKD device shares both the generated key ID and the key with the initiating router.
5. Key ID sharing with peer: The initiating router shares only the key ID with its peer router.
6. Key retrieval by peer: The peer router uses the received key ID to retrieve the corresponding encryption key from its own QKD device.
7. Secure link establishment: Both routers now possess the same MACsec encryption key, enabling them to establish the secure point-to-point MACsec link.

Result

Quantum networks securely communicate encryption keys. This enables robust and automated secure communication links between peer router interfaces. Routers do not directly exchange sensitive encryption keys.

Restrictions for MACsec encryption using SKIP

Before implementing MACsec encryption using the SKIP protocol, you must consider the following restrictions:

- Use the SKIP protocol only on 8202-32FH-M routers.
- Configure SKIP only for point-to-point MACsec encryption.
- Enable SKIP protocol only on interfaces that support MACsec encryption.

Configure point-to-point MACsec encryption using SKIP

Establish secure, point-to-point MACsec encryption between two routers using the SKIP protocol and Quantum Key Distribution (QKD) for automated, quantum-safe key management.

Use this task when you need to configure MACsec in Pre-placed Key (PPK) mode with keys provided by external QKD devices and SKIP for secure key provisioning. This enhances security by leveraging quantum key exchange for MACsec.

Before you begin

- Configure MACsec Pre-Shared Key (PSK). For more information, see Configure a MACsec keychain, on page 12.
- Configure MACsec in the PPK mode.
- Ensure that you have a network of external QKD devices.
- Add the QKD server CA to the trustpoint in the router. For more information, see *Configure Trustpoint* section in the *System Security Configuration Guide for Cisco 8000 Series Routers*.
- Import the QKD server root CA certificate in the router. For more information, see *Configure Certificate Enrollment Using Cut-and-Paste* section in the *System Security Configuration Guide for Cisco 8000 Series Routers*.

Procedure

Step 1 Configure the QKD profile.

a) On Router 1, enter global configuration mode, define the SKS profile, and specify the remote KME server:

Example:

```
Router# config
Router(config)# sks profile ProfileR1toR2 type remote
```

```
Router(config-sks-profile)# kme server ipv4 192.0.2.34 port 10001
Router(config-sks-profile)# commit
```

b) On Router 2, enter global configuration mode, define the SKS profile, and specify the remote KME server:

Example:

```
Router# config
Router(config)# sks profile ProfileR2toR1 type remote
Router(config-sks-profile)# kme server ipv4 192.0.2.35 port 10001
Router(config-sks-profile)# commit
```

Step 2 Map the QKD profile to the MACsec policy.

a) On Router 1:

Example:

```
Router# config
Router(config)# macsec-policy R1toR2
Router(config-macsec-policy)# ppk sks-profile ProfileR1toR2
Router(config-macsec-policy)# commit
```

b) On Router 2:

Example:

```
Router# config
Router(config)# macsec-policy R2toR1
Router(config-macsec-policy)# ppk sks-profile ProfileR2toR1
Router(config-macsec-policy)# commit
```

Step 3 Apply MACsec policy to the interfaces.

a) On Router 1:

Example:

```
Router# config
Router(config)# interface hundredGigE 0/1/0/10
Router(config-if)# ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R1toR2
Router(config)# commit

Router(config)# interface hundredGigE 0/1/0/11
Router(config-if)# ipv4 address 192.0.3.1 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R1toR2
Router(config)# commit

Router(config)# interface hundredGigE 0/1/0/12
Router(config-if)# ipv4 address 192.0.4.1 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R1toR2
Router(config)# commit

Router(config)# interface hundredGigE 0/1/0/9
Router(config-if)# ipv4 address 192.0.5.1 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R1toR2
Router(config)# commit
```

b) On Router 2:

Example:

```
Router# config
Router(config)# interface hundredGigE 0/1/0/10
Router(config-if)# ipv4 address 192.0.2.2 255.255.255.0
```

```
Router(config-if)# macsec psk-keychain mac_chain policy R2toR1
Router(config-if)# commit

Router(config)# interface hundredGigE 0/1/0/11
Router(config-if)# ipv4 address 192.0.3.2 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R2toR1
Router(config-if)# commit

Router(config)# interface hundredGigE 0/1/0/12
Router(config-if)# ipv4 address 192.0.4.2 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R2toR1
Router(config-if)# commit

Router(config)# interface hundredGigE 0/1/0/9
Router(config-if)# ipv4 address 192.0.5.2 255.255.255.0
Router(config-if)# macsec psk-keychain mac_chain policy R2toR1
Router(config-if)# commit
```

Step 4 Verify the configurations in each router using the **show running config** command.

a) On Router 1:

Example:

```
sks profile ProfileR1toR2 type remote
 kme server ipv4 192.0.2.34 port 10001
!
macsec-policy R1toR2
 ppk
  sks-profile ProfileR1toR2
 !
!
interface hundredGigE 0/1/0/10
 ipv4 address 192.0.2.1 255.255.255.0
 macsec psk-keychain mac_chain policy R1toR2
!
interface hundredGigE 0/1/0/11
 ipv4 address 192.0.3.1 255.255.255.0
 macsec psk-keychain mac_chain policy R1toR2
!
interface hundredGigE 0/1/0/12
 ipv4 address 192.0.4.1 255.255.255.0
 macsec psk-keychain mac_chain policy R1toR2
!
interface hundredGigE 0/1/0/9
 ipv4 address 192.0.5.1 255.255.255.0
 macsec psk-keychain mac_chain policy R1toR2
!
```

b) On Router 2:

Example:

```
sks profile ProfileR2toR1 type remote
 kme server ipv4 192.0.2.35 port 10001
!
macsec-policy R2toR1
 ppk
  sks-profile ProfileR2toR1
 !
!
interface hundredGigE 0/1/0/10
 ipv4 address 192.0.2.2 255.255.255.0
 macsec psk-keychain mac_chain policy R2toR1
!
```

```
interface hundredGigE 0/1/0/11
 ipv4 address 192.0.3.2 255.255.255.0
 macsec psk-keychain mac_chain policy R2toR1
!
interface hundredGigE 0/1/0/12
 ipv4 address 192.0.4.2 255.255.255.0
 macsec psk-keychain mac_chain policy R2toR1
!
interface hundredGigE 0/1/0/9
 ipv4 address 192.0.5.2 255.255.255.0
 macsec psk-keychain mac_chain policy R2toR1
!
```

Step 5 Verify the point-to-point MACsec encryption using SKIP on either router with the **show crypto sks profile all** and **show crypto sks profile all** commands.

Example:

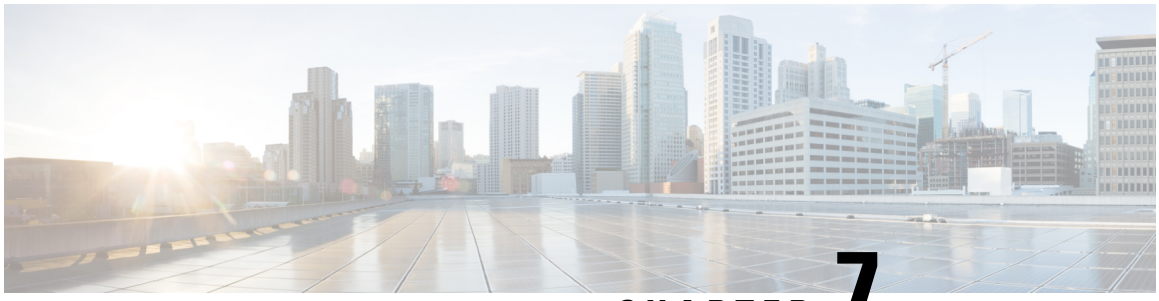
```
Router(ios)# show crypto sks profile all
Profile Name         :ProfileR1toR2
Myidentifier        :Router1
Type                :Remote
Reg Client Count     :1

Server
IP                   :192.0.2.34
Port                 :10001
Vrf                  :Notconfigured
Source Interface     :Notconfigured
Status               :Connected
Entropy              :true
Key                  :true
Algorithm            :QKD
Local identifier     :Alice
Remote identifier    :Alice

Peerlist
QKD ID               :Bob
State                :Connected

Peerlist
QKD ID               :Alice
State                :Connected


Router(ios)# show crypto sks profile all stats
Profile Name            : ProfileR1toR2
My identifier           : Router1
Server
 IP                     : 192.0.2.34
 Port                   : 10001
 Status                 : connected
Counters
 Capability request         : 1
 Key request                : 3
 Key-id request             : 0
 Entropy request            : 0
 Capability response        : 1
 Key response               : 3
 Key-id response            : 0
 Entropy response           : 0
 Total request              : 4
 Request failed             : 0
 Request success            : 4
```

```
 Total response             : 4
 Response failed            : 0
 Response success           : 4
 Retry count                : 0
 Response Ignored           : 0
 Cancelled count            : 0
Response time
 Max Time                   : 100 ms
 Avg Time                   : 10  ms
 Min Time                   : 50  ms
Last transaction
 Transaction Id             : 9
 Transaction type           : Get key
 Transaction status         : Response data received, successfully
 Http code                  : 200 OK (200)
```

When the task is completed, MACsec link encryption is established between both routers using SKIP and QKD for secure key provisioning. All interfaces configured with the MACsec policy exchange encrypted and authenticated traffic.

What to do next

Monitor SKS profile status and key exchange statistics to confirm ongoing secure operation. Review logs and counters for negotiation failures or changes in link state.



CHAPTER 7

Secure MACsec encryption

This chapter provides detailed guidance on securing MACsec-enabled routers, including configuring Power-on Self-Test (KAT) for FIPS compliance, managing dynamic power allocation for MACsec ports, and implementing secure MACsec pre-shared keys using Type 6 password encryption. Users can follow step-by-step procedures to ensure cryptographic integrity, robust key management, and optimal power distribution on supported routers.

Power-on Self-Test KAT for Common Criteria and FIPS

A power-on self-test (POST) is a security mechanism that

- verifies the cryptographic integrity of hardware components at system startup,
- prevents network traffic flow if integrity checks fail, and
- supports compliance with security standards such as Common Criteria and FIPS.

Power-on self-tests utilize Known Answer Tests (KATs) executed immediately after powering on the cipher module in MACsec-enabled Cisco 8000 series routers. These tests check cryptographic algorithms (e.g., SHA, DES) on each physical layer chip (PHY) with hardware crypto. If any PHY fails the test, the module enters an error state and does not allow traffic, ensuring only secure, verified hardware is operational.

The POST KAT feature is now available on Cisco 8800 48x100 GbE QSFP28 Line Card (8800-LC-48H), Cisco 8800 36x400GE QSFP56-DD Line Card with MACsec (8800-LC-36FH-M), and Cisco 8606 series routers.

- On successful POST KAT execution, the system displays logs indicating KAT Test PASSED for each port, and the corresponding line card becomes operational.
- If POST KAT fails on any PHY, the system logs a KAT Test FAILED message, the line card enters an ERROR state, and network traffic is blocked on that card.

How MACsec pre-shared keys with Type 6 password encryption work

Summary

MACsec pre-shared keys with Type 6 password encryption safeguard Layer 2 traffic by encrypting cryptographic secrets at rest using an AES-based method linked to a local primary password-encryption key. The router encrypts PSKs for storage and decrypts them in memory only when required for MACsec operations, ensuring secure handling and rotation of secrets.

The key components involved in the process are:

- Primary password-encryption key: A locally defined secret serving as the root key for securing all Type 6–encrypted strings on the router.
- Type 6 encryption engine: An AES-based service that encrypts and decrypts secret values for secure storage and controlled display.
- MACsec PSK entries: The pre-shared keys used by MACsec; stored as encrypted strings when Type 6 is enabled.
- Configuration datastore: The running and startup configuration repositories that persist encrypted secret strings.
- Router: Hosts the primary key, runs the Type 6 engine, encrypts and decrypts PSKs, and manages configuration persistence.
- User: Defines the primary key, enables Type 6 encryption, configures MACsec PSKs, and performs key rotation.

Workflow

These are the stages MACsec pre-shared keys with Type 6 password encryption:

1. Primary key setup: The administrator sets a primary password-encryption key that meets policy requirements. The router internally stores it and uses it as the root to protect all Type 6–encrypted strings.
2. Enable Type 6 encryption: The administrator activates the AES-based Type 6 mechanism. The router binds the encryption engine to the primary key, ensuring new or updated secret strings are encrypted at rest.
3. Enter PSKs and store securely: When MACsec PSKs are configured or modified, the router accepts plaintext input, encrypts it immediately using Type 6 with the primary key, and saves only the encrypted representation in the configuration.
4. Use PSKs for MACsec: When MACsec needs a PSK, the router decrypts the stored Type 6 value in memory using the primary key and provides the plaintext to MACsec to establish secure sessions.
5. Rotate the primary key (optional): When the primary key is rotated, the router re-encrypts all existing Type 6 strings, including MACsec PSKs, under the new key after the administrator authenticates and supplies the new key.

Result

The process ensures that MACsec PSKs remain encrypted at rest, prevents plaintext exposure in configurations, and supports controlled key rotation, thus enhancing the security posture of Layer 2 traffic protection.

Guidelines for MACsec FIPS-POST and KAT

Expect boot-up delays

Expect a boot-up delay of approximately 2 to 3 minutes for a line card when you enable Known Answer Test (KAT) compared to when it is not enabled.

Prevent configuration conflicts

Ensure that if Power-On Self-Test (POST) Known Answer Test (KAT) is already enabled on the PHY, you do not configure the **hw-module macsec-fips-post location all** command again. This prevents configuration conflicts, especially during a configuration restore. Use the **show hw-module macsec-mode fips-post** command to view the current running configurations in such scenarios.

Enable Power-on Self-Test KAT for MACsec FIPS cards

Ensure MACsec FIPS line cards on routers conduct Power-on Self-Test Known Answer Tests (KAT) to verify cryptographic integrity and support FIPS compliance.

This task is essential when deploying or maintaining routers with MACsec FIPS line cards to confirm hardware cryptographic integrity.

KAT is not enabled by default. You can configure the `hw-module macsec-fips-post` command to enable POST KAT for the MACsec-enabled hardware. With this configuration, the KAT always runs as a self-test during power on. The cryptographic algorithm tests are performed on every physical layer chip (PHY) with hardware crypto once it is powered up.

- Pass criteria for KAT: Any change in the FIPS mode configuration requires a line card reload. On reload, the FIPS POST is run as part of the line card boot sequence. The subsequent boot (based on the FIPS mode) state re-triggers the KAT. If there are multiple PHYs hardware in a module, the system performs the KAT on each PHY and returns the KAT results. If all PHYs pass the KAT, the system brings up the line card for regular usage.
- Fail criteria for KAT: Traffic does not pass through a MACsec-enabled PID that failed KAT. If any PHY registers a KAT failure, the module enters an ERROR state and the system displays a critical ERROR SYSLOG output: `KAT Test Failed`. The system does not allow any traffic or data flow through the interfaces on that line card. Although the interfaces are present, they do not come up or allow traffic to flow through them on a line card that failed KAT. In a modular chassis, all other line cards, except the one that failed the KAT, will be up and running.

Before you begin

- Install the k9sec package on the router.
- Confirm that FIPS is supported and enabled on the line card.

Follow these steps to enable and verify Power-on Self-Test KAT for MACsec FIPS cards:

Procedure

Step 1 Use the **hw-module macsec-fips-post** command to configure the Power-on Self-Test KAT on the desired line card.

Example:

```
Router#config
Router(config)#hw-module macsec-fips-post location 0/4/CPU0
Router(config)#commit
```

Step 2 Use the **show hw-module macsec-fips-post** command to verify the Power-on Self-Test KAT on a line card.

Example:

Before configuring POST KAT:

```
Router#show hw-module macsec-fips-post
Wed Jun 17 09:29:18.780 UTC

Location       Configured     Applied          Action
--------------------------------------------------------------
0/0/CPU0       NO             NO               NONE          >>> LC36
0/11/CPU0      NO             NO               NONE          >>> LC48
```

After configuring the command for POST KAT, and before the line card reload:

```
Router#show hw-module macsec-fips-post
Wed Jun 17 09:36:31.932 UTC

Location       Configured     Applied          Action
--------------------------------------------------------------
0/0/CPU0       NO             NO               NONE
0/11/CPU0      YES            NO               RELOAD
```

After the line card reload:

```
Router#show hw-module macsec-fips-post
Wed Jun 17 10:03:57.263 UTC

Location       Configured     Applied          Action
--------------------------------------------------------------
0/0/CPU0       NO             NO               NONE
0/11/CPU0      YES            YES              NONE
```

Step 3 Review system logs to verify results for KAT execution on each port.

Example:

These are sample logs displayed after a successful KAT. The system performs KAT on each port, but the ports may not be in order in the display output.

```
Router#show logging | inc KAT
Wed Jun 10 12:07:29.849 UTC
LC/0/4/CPU0:Jun 9 10:37:37.521 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 0
LC/0/4/CPU0:Jun 9 10:37:37.522 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 28
LC/0/4/CPU0:Jun 9 10:37:37.522 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 27
LC/0/4/CPU0:Jun 9 10:37:37.522 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 1
LC/0/4/CPU0:Jun 9 10:39:10.393 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 2
```

```
LC/0/4/CPU0:Jun 9 10:39:10.393 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 6
LC/0/4/CPU0:Jun 9 10:39:10.393 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 7
LC/0/4/CPU0:Jun 9 10:39:10.393 UTC: optics_driver[159]: %L2-SECY_DRIVER-6-KAT_PASS : KAT Test PASSED
 for Port No: 8
```

These are sample logs displayed in KAT failure scenarios:

```
Router#show logging | inc SECY
Thu Jul 16 09:13:29.217 UTC
LC/0/7/CPU0:Jul 16 08:41:30.709 UTC: optics_driver[152]: %L2-SECY_DRIVER-0-KAT_FAIL_DETECTED : KAT
Test FAILED for Port No: 0
LC/0/7/CPU0:Jul 16 08:41:30.709 UTC: optics_driver[152]: %L2-SECY_DRIVER-0-KAT_FAIL_DETECTED : KAT
Test FAILED for Port No: 47
LC/0/7/CPU0:Jul 16 08:41:30.709 UTC: optics_driver[152]: %L2-SECY_DRIVER-0-KAT_FAIL_DETECTED : KAT
Test FAILED for Port No: 7
LC/0/7/CPU0:Jul 16 08:41:30.709 UTC: optics_driver[152]: %L2-SECY_DRIVER-0-KAT_FAIL_DETECTED : KAT
Test FAILED for Port No: 6
```

MACsec FIPS line cards run Power-on Self-Test KAT upon reload. Successful PASS results are logged for each port; failures are flagged for further troubleshooting.

What to do next

If any port reports KAT FAIL, investigate and resolve hardware or configuration issues before continuing with production use.

Dynamic power management for MACsec-enabled ports

Dynamic Power Management for MACsec-enabled ports is a MACsec function that

- allocates total power to a router and its fabric or line cards based on various factors,
- validates power availability for MACsec sessions on configured interfaces, and
- prevents MACSec sessions from establishing if power is insufficient.

The dynamic power management feature distributes total available power to a router and its fabric cards or line cards based on factors such as the number and type of cards installed, their operating modes, card combinations, NPU (Network Processing Unit) power mode, and optics. When MACSec is configured on interfaces, the software checks internally if there is enough power to bring up all intended MACSec sessions. If the system cannot power all configured MACSec sessions, some sessions remain down regardless of the interface configuration.

When this situation occurs, the router console logs a message indicating the reason. Users can remove MACSec configurations from affected interfaces or add more Power Supply Units (PSUs) to meet new power requirements. If MACSec configurations remain on downed sessions, those sessions are not guaranteed to recover after a router or line card reload.

The router console displays a log message in such cases, indicating the reason for session failure. Users can choose to remove the MACSec configuration from the corresponding interfaces or re-provision the Power Supply Units (PSUs) based on the additional power requirement for new sessions. If MACSec configurations

are not removed for sessions that are down, there is no guarantee that the same MACSec sessions that were brought up earlier will come up after a router or line card reload.

By default, dynamic power management is enabled. You can disable it using the following command in XR Config mode: **no power-mgmt action**.

If insufficient power is available for MACSec sessions, you might see a log message such as:

```
LC/0/4/CPU0:Dec 21 07:35:27.977 UTC: macsec_mka[131]:
%L2-MKA-5-MACSEC_POWER_STATUS_ERR : (Hu0/4/0/9), Insufficient power
```

Hardware support matrix for dynamic power management for MACsec-enabled ports

Cisco IOS XR Software Release	Product ID
Release 25.1.1	8712-MOD-M
Release 24.4.1	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM 8212-48FH-M 8711-32FH-M
Release 7.3.3	88-LC0-36FH-M 88-LC0-34H14FH 8800-LC-48H

Verify dynamic power management for MACSec-enabled ports

Confirm that power is correctly allocated and released for MACSec-enabled interfaces and that chassis and component power levels are appropriate.

Use this task to monitor and verify power allocation for MACSec interfaces on Cisco routers. This includes checking syslog messages, reviewing chassis and line card power usage, and confirming the MACSec power status at the interface level.

Procedure

Step 1 Monitor syslog messages for power allocation and release events for MACSec interfaces.

- When power is allocated to a MACSec interface, expect a syslog entry similar to:

  ```
  LC/slot/CPU: macsec_mka: %L2-MKA-5-MACSEC_POWER_STATUS : (interface), Power allocated
  ```

- When power is released (such as when MACSec policy is removed), expect a syslog entry similar to:

  ```
  LC/slot/CPU: macsec_mka: %L2-MKA-5-MACSEC_POWER_STATUS : (interface), Power released
  ```

Step 2 Use the **show environment power** command to review chassis-level power information.

Example:

```
Router# show environment power
Thu Dec  9 11:12:54.239 UTC
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================
   Total output power capacity (N + 1)             :   31500W +     6300W
   Total output power required                     :   11208W
   Total power input                               :    3778W
   Total power output                              :    3395W

===============================================================================
   Power       Supply         -------Input--------   -----Output---     Status
   Module      Type            Volts A/B   Amps A/B   Volts     Amps
===============================================================================
   0/PT0-PM0   PSU6.3KW-HV     246.0/244.3 1.2/1.2    55.3      9.9      OK
   0/PT0-PM1   PSU6.3KW-HV     245.7/244.3 1.3/1.3    55.4      10.1     OK
   0/PT0-PM2   PSU6.3KW-HV     245.7/246.3 1.5/1.2    55.4      10.3     OK
   0/PT1-PM0   PSU6.3KW-HV     246.0/246.0 1.3/1.3    55.4      10.3     OK
   0/PT1-PM1   PSU6.3KW-HV     244.3/244.6 1.3/1.3    55.1      10.7     OK
   0/PT1-PM2   PSU6.3KW-HV     245.7/245.5 1.3/1.2    55.2      10.1     OK
   0/PT2-PM0   PSU6.3KW-HV     0.0/0.0     0.0/0.0    0.0       0.0      FAILED or NO PWR
   0/PT2-PM1   PSU6.3KW-HV     0.0/0.0     0.0/0.0    0.0       0.0      FAILED or NO PWR
   0/PT2-PM2   PWR-6.3KW-HV    0.0/0.0     0.0/0.0    0.0       0.0      FAILED or NO PWR

Total of Power Modules:       3778W/15.4A              3395W/61.4A

===============================================================================
   Location     Card Type               Power       Power        Status
                                                    Allocated    Used
                                                    Watts        Watts
===============================================================================
   0/RP0/CPU0   8800-RP-O               95          78           ON
   0/RP1/CPU0   8800-RP-O               95          -            ON
   0/0/CPU0     88-LC0-36FH-O           934         543          ON
   0/1/CPU0      -                      102         -            RESERVED
   0/2/CPU0     8800-LC-48H-O           778         474          ON
   0/3/CPU0     -                       102         -            RESERVED
   0/4/CPU0     -                       102         -            RESERVED
   0/5/CPU0     -                       102         -            RESERVED
   0/6/CPU0     8800-LC-48H             102         -            OFF
   0/7/CPU0     -                       102         -            RESERVED
   0/8/CPU0     -                       102         -            RESERVED
   0/9/CPU0     -                       102         -            RESERVED
   0/10/CPU0    -                       102         -            RESERVED
   0/11/CPU0    -                       102         -            RESERVED
   0/FC0        -                       26          -            RESERVED
   0/FC1        8812-FC                 784         338          ON
   0/FC2        8812-FC                 784         337          ON
   0/FC3        8812-FC                 784         343          ON
   0/FC4        8812-FC                 784         338          ON
   0/FC5        8812-FC                 784         344          ON
   0/FC6        -                       26          -            RESERVED
   0/FC7        -                       26          -            RESERVED
   0/FT0        SF-D-12-FAN             1072        135          ON
   0/FT1        SF-D-12-FAN             1072        105          ON
   0/FT2        SF-D-12-FAN             1072        123          ON
   0/FT3        SF-D-12-FAN             1072        123          ON
```

Verify total output power capacity, required power, input/output levels, and status of each power module.

Step 3 Use the **show environment power allocated location** command to verify power allocated for each component on a line card.

Example:

```
Router# show environment power allocated location 0/2/CPU0
Thu Dec  9 09:53:49.921 UTC
================================================================================
   Location    Components               Power
                                        Allocated
                                        Watts
================================================================================
  0/2/CPU0     Data-path                772
               MACSEC                     3
               OPTICS                     3
================================================================================
               Total                    778
```

Confirm that the appropriate wattage is allocated for the MACSec component on each relevant line card.

Step 4 Use the **show environment power allocated details location** command to see interface-level power allocation.

Example:

```
Router# show environment power allocated details location 0/2/CPU0
Thu Dec  9 09:53:49.921 UTC
================================================================================
   Location    Components               Power
                                        Allocated
                                        Watts
================================================================================
  0/2/CPU0     Data-path                772
               0/2/0/9                  3
               0/2/0/0                    3
================================================================================
               Total                    778
```

Verify that the correct power is allocated for MACSec on each specific interface where MACSec is enabled.

Step 5 Use the **show macsec mka interface detail** command to verify MACSec power status at the interface level.

Example:

```
Router# show macsec mka interface hundredGigE 0/2/0/9 detail
Tue Dec 21 08:10:41.571 UTC
Interface Name : HundredGigE0/2/0/9
 Interface Namestring : HundredGigE0/2/0/9
 Interface short name : Hu0/2/0/9
 Interface handle : 0x2000480
 Interface number : 0x2000480
 MacSecControlledIfh : 0x20005b8
 MacSecUnControlledIfh : 0x20005c0
 Interface MAC : 34ed.1b5b.d047
 Ethertype : 888E
 EAPoL Destination Addr : 0180.c200.0003
 MACsec Shutdown : FALSE
 Config Received : TRUE
 IM notify Complete : TRUE
 MACsec Power Status : Allocated
 Interface CAPS Add : TRUE
 RxSA CAPS Add : TRUE
 TxSA CAPS Add : TRUE
 MKA PSK Info
  Key Chain Name : psk
  MKA Cipher Suite : AES-128-CMAC
  CKN : 22 22
 MKA fallback_PSK Info
  fallback keychain Name : - NA -
 Policy : p3
```

Confirm that the MACsec Power Status field shows Allocated for interfaces with MACSec enabled.

Power is appropriately allocated or released for MACSec-enabled ports. Syslog entries confirm power status changes, and show commands verify that power is provisioned and reported as expected at the chassis, line card, and interface levels.

MACsec pre-shared keys with Type 6 password encryption

A MACsec pre-shared key with Type 6 password encryption is a router security configuration that

- securely stores MACsec Connectivity Association Keys (CAKs) in encrypted form,
- depends on a locally configured primary key to operate, and
- uses AES-256 symmetric encryption to protect MACsec key material in the router configuration.
- Primary key: The local password or key the router uses to encrypt and decrypt all MACsec CAKs stored in configuration. The device does not save this key in configuration and it is not viewable.
- Type 6 password encryption: A Cisco encryption scheme that applies AES-256 symmetric encryption to sensitive secrets in configuration, enabling the system to decrypt on demand to establish secure communication.
- MACsec CAK / PSK: The static pre-shared key MACsec uses to form a Connectivity Association between peers.

When enabled, the PSK does not appear in clear text in running, startup, or archived configurations; the router stores only an encrypted value that it can decrypt locally when needed. Type 6 password encryption functions only when a primary key is configured.

Benefits of securing MACsec pre-shared keys with Type 6 password encryption

- Protects MACsec PSKs from exposure in plain text.
- Utilizes AES-256 encryption for robust and modern cryptographic protection.
- Supports compliance with regulatory and organizational security policies.
- Reduces insider threat risks from configuration file inspection.

Configure MACsec pre-shared keys with Type 6 password encryption

Configure MACsec pre-shared keys with Type 6 encrypted passwords for secure key management.

Perform this task to set up or modify MACsec PSK with Type 6 password encryption.

Procedure

Step 1 Use the **key config-key password-encryption** command to create the primary key.

Example:

```
Router# config
Router(config)# key config-key password-encryption
Enter new key:
Enter confirm key:
Router(config)# commit
```

- When prompted, set a new password with the following requirements:
  - Minimum length: 6 characters
  - Maximum length: 64 characters
  - Allowed characters: uppercase letters [A-Z], lowercase letters [a-z], and digits [0-9]

Step 2 Use the **key chain** command to configure the macsec keychain.

Example:

```
Router# config
Router(config)# key chain kc1 macsec
Router(config-kc1-MacSec)# key 1111
Router(config-kc1-MacSec-1111)# key-string
12345678901234567890123456789022123456789012345678901234567890123456789022 cryptographic-algorithm aes-256-cmac
Router(config-kc1-MacSec-1111)# lifetime 00:00:00 1 October 2019 infinite
Router(config-kc1-MacSec-1111)# commit
```

Modify the primary key if needed:

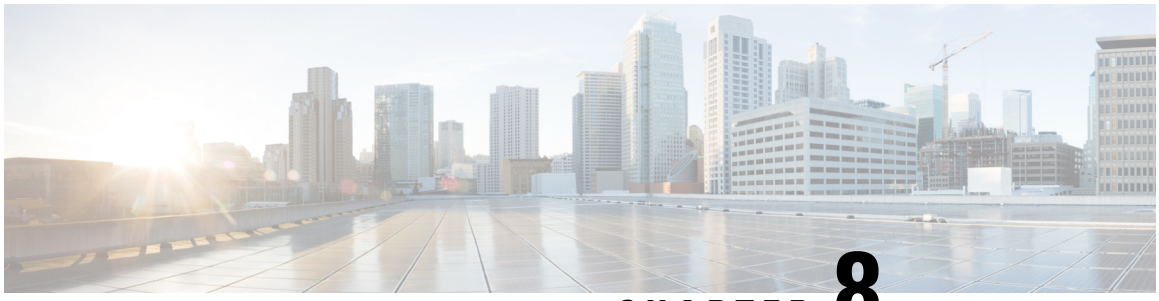
- If a primary key exists, enter the current key when prompted before setting a new key.
- Modifying the primary key re-encrypts all existing Type 6 key strings with the new key.
- Ensure the `password6 encryption aes` command is configured to enable re-encryption; otherwise, the update will fail.

Primary key deletion will bring down MACsec traffic if MKA sessions are up with Type 6 keys. To avoid traffic disruptions, configure a new set of PSK key pairs [key (CKN) and key string (CAK)] with latest timestamps with the lifetime of infinite validity on both the peers and ensure the successful CAK rekey to the newly configured CKN and CAK.

Delete the primary key when necessary:

```
Router# config
Router(config)# no password6 encryption aes
Router(config)# commit
Router(config)# exit
Router# key config-key password-encryption delete
```

The primary key and Type 6 password encryption are successfully configured, modified, or deleted, and the MACsec key chain is configured with Type 6 encrypted pre-shared keys, ensuring secure key management.



CHAPTER 8

MACsec encryption performance and statistics

This chapter provides comprehensive guidance on monitoring and troubleshooting MACsec performance using SecY statistics, SNMP MIBs, and CLI commands. Users can learn how to access detailed encryption and decryption metrics, retrieve MACsec interface indexes, and perform SNMP queries for secure network management and diagnostics.

MACsec SecY statistics

The MACsec SecY statistics are operational metrics that

- monitor the performance of the MAC Security (MACsec) Secure Channel (SecY) component,
- provide detailed visibility into packet and octet processing activities, and
- help identify encryption or decryption issues in secure network communication.

MACsec SecY statistics track the behavior of encrypted traffic, including packet processing, encryption, decryption, and error conditions. They serve as diagnostic indicators that allow network administrators to confirm proper MACsec operation and troubleshoot encrypted traffic flows.

Key aspects of SecY statistics include:

- Interface statistics: Track untagged packets, packets without MACsec tags, packets with invalid tags, unknown Secure Channel Identifiers (SCI), and counts of validated or decrypted octets.
- Secure Channel (SC) statistics: Include transmit (TxSC) and receive (RxSC) data, such as packets protected, encrypted, dropped for being too long, and octet encryption or decryption counts.
- Secure Association (SA) statistics: Provide detailed per-SA data for both transmit and receive directions, including packets protected, encrypted, and the next packet number (NextPN).

These statistics can be accessed using CLI commands such as **show macsec secy stats** on supported controllers or interfaces, and through SNMP queries using the IEEE8021-SECY-MIB.

Network administrators rely on these statistics to ensure that MACsec encryption is functioning correctly and to detect anomalies in encrypted traffic.

Administrators can query MACsec SecY statistics using the following methods:

- CLI – for real-time interface and controller-level statistics
- SNMP MIB – for remote monitoring and integration with network management systems

Query SNMP statistics

Administrators can query SNMP statistics through the CLI to view detailed information about MACsec SecY statistics on a specific interface.

Use the **show macsec secy statistics interface** command to display detailed MACsec SecY statistics for a specified interface.

- Example:

```
Router# show macsec secy stats interface hundredGigE 0/1/0/10 sc

Interface Stats
    InPktsUntagged     :  0
    InPktsNoTag        :  0
    InPktsBadTag       :  0
    InPktsUnknownSCI   :  0
    InPktsNoSCI        :  0
    InPktsOverrun      :  0
    InOctetsValidated  :  0
    InOctetsDecrypted  :  0
    OutPktsUntagged    :  0
    OutPktsTooLong     :  0
    OutOctetsProtected :  0
    OutOctetsEncrypted :  0

SC Stats
  TxSC Stats
    OutPktsProtected   : 0
    OutPktsEncrypted   : 0
    OutOctetsProtected : 0
    OutOctetsEncrypted : 0
    OutPktsTooLong     : 0
    TxSA Stats
      TxSA 0:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 1
      TxSA 1:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0
      TxSA 2:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0
      TxSA 3:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0

  RxSC Stats
    RxSC 1: 10000742d968a00
```

```
    InPktsUnchecked     : 0
    InPktsDelayed       : 0
    InPktsLate          : 0
    InPktsOK            : 0
    InPktsInvalid       : 0
    InPktsNotValid      : 0
    InPktsNotUsingSA    : 0
    InPktsUnusedSA      : 0
    InPktsUntaggedHit   : 0
    InOctetsValidated   : 0
    InOctetsDecrypted   : 0
  RxSA Stats
    RxSA 0:
      InPktsUnusedSA      : 0
      InPktsNotUsingSA    : 0
      InPktsNotValid      : 0
      InPktsInvalid       : 0
      InPktsOK            : 0
      NextPN              : 1
    RxSA 1:
      InPktsUnusedSA      : 0
      InPktsNotUsingSA    : 0
      InPktsNotValid      : 0
      InPktsInvalid       : 0
      InPktsOK            : 0
      NextPN              : 0
    RxSA 2:
      InPktsUnusedSA      : 0
      InPktsNotUsingSA    : 0
      InPktsNotValid      : 0
      InPktsInvalid       : 0
      InPktsOK            : 0
      NextPN              : 0
    RxSA 3:
      InPktsUnusedSA      : 0
      InPktsNotUsingSA    : 0
      InPktsNotValid      : 0
      InPktsInvalid       : 0
      InPktsOK            : 0
      NextPN              : 0
```

- On Cisco 8712-MOD-M routers, all TxSC (Transmit Secure Channel) counters display a value of zero. This behavior occurs due to a hardware limitation — K100 ASIC-based systems used in these routers do not support the collection of TxSC statistics.

MACsec SNMP MIB

A MACsec SNMP MIB (IEEE8021-SECY-MIB) is a management information base that

- provides Simple Network Management Protocol (SNMP) access to the MAC Security (MACsec) entity (SecY),
- enables network administrators to query encryption, decryption, and hardware-related SecY data, and
- operates exclusively on the Controlled Port for MACsec-enabled interfaces.

The IEEE8021-SECY-MIB allows monitoring of SecY statistics on IOS XR MACsec-enabled line cards, offering visibility into the performance and status of secure data transmission. It is primarily used to retrieve real-time operational data about packet encryption and decryption within MACsec environments.

The object identifier (OID) for the IEEE8021-SECY-MIB is 1.0.8802.1.1.3.

The IEEE8021-SECY-MIB contains the following tables that specifies the detailed attributes of the MACsec Controlled Port interface index.

Table 11: IEEE8021-SECY-MIB

Tables	OID
secyIfTable	1.0.8802.1.1.3.1.1.1
secyTxSCTable	1.0.8802.1.1.3.1.1.2
secyTxSATable	1.0.8802.1.1.3.1.1.3
secyRxSCTable	1.0.8802.1.1.3.1.1.4
secyRxSATable	1.0.8802.1.1.3.1.1.5
secyCipherSuiteTable	1.0.8802.1.1.3.1.1.6
secyTxSAStatsTable	1.0.8802.1.1.3.1.2.1
secyTxSCStatsTable	1.0.8802.1.1.3.1.2.2
secyRxSAStatsTable	1.0.8802.1.1.3.1.2.3
secyRxSCStatsTable	1.0.8802.1.1.3.1.2.4
secyStatsTable	1.0.8802.1.1.3.1.2.5

- For more technical details on MACsec SNMP MIB (IEEE8021-SECY-MIB), download the IEEE8021-SECY-MIB from MIB Locator in Cisco Feature Navigator.
- For more information on the IEEE8021-SecY-MIB, see http://www.ieee802.org/1/files/public/MIBs/IEEE8021-SECY-MIB-200601100000Z.mib

Use SNMP commands to access SECY MIB

Retrieve SECY MIB information from a device using SNMP commands.

You need to query SECY MIB data for MACsec interfaces on a device with SNMP enabled.

Before you begin

Ensure you have the correct SNMP community string, management IP address, and interface ifIndex.

Follow these steps to retrieve SECY MIB data:

Procedure

Step 1 Walk the entire SECY MIB subtree to enumerate all objects.

Example:

```
snmpwalk -v2c -c <community_string> <management_IP> 1.0.8802.1.1.3
```

Step 2 Query the TxSCI value for a specific interface using its ifIndex:

Example:

```
snmpget -v2c -c <community_string> <management_IP> iso.0.8802.1.1.3.1.1.2.1.1.<ifIndex>
```

Step 3 Find the ifIndex of the MACsec controlled port by performing an SNMP walk on the IfMib:

Example:

```
snmpwalk -v2c -c <community_string> <management_IP> 1.3.6.1.2.1.31.1.1.1.1
```

Step 4 Alternatively, use the **show snmp interface** command to display SNMP interface information:

You will obtain SECY MIB data and the interface index needed for targeted SNMP queries.

Obtain the MACsec controlled port interface index

This reference describes how to identify the interface index (ifindex) for a MACsec controlled port by using SNMP and CLI commands. It helps users manage and monitor MACsec-enabled interfaces on network devices.

Use these commands to obtain the ifindex of the MACsec controlled port:

- **snmpwalk** command on IfMib [OID: 1.3.6.1.2.1.31.1.1.1]

```
rtr1.0/1/CPU0/ $ snmpwalk -v2c -c public 10.0.0.1  1.3.6.1.2.1.31.1.1.1.1
SNMPv2-SMI::mib-2.31.1.1.1.1.3 = STRING: "GigabitEthernet0/1/0/0"
SNMPv2-SMI::mib-2.31.1.1.1.1.18 = STRING: "MACSecControlled0/1/0/0"
SNMPv2-SMI::mib-2.31.1.1.1.1.19 = STRING: "MACSecUncontrolled0/1/0/0"
```

- **show snmp interface** command

```
Router# show snmp interface
.
.
ifName : MACSecControlled0/0/0/0  ifIndex : 77
ifName : MACSecControlled0/0/0/4  ifIndex : 79
ifName : MACSecControlled0/0/0/21  ifIndex : 94
ifName : MACSecControlled0/0/0/30  ifIndex : 118
ifName : MACSecControlled0/0/0/34  ifIndex : 116
ifName : MACSecUncontrolled0/0/0/0  ifIndex : 78
ifName : MACSecUncontrolled0/0/0/4  ifIndex : 80
ifName : MACSecUncontrolled0/0/0/21  ifIndex : 95
ifName : MACSecUncontrolled0/0/0/30  ifIndex : 119
ifName : MACSecUncontrolled0/0/0/34  ifIndex : 117
```

SNMP query examples

The following commands enable network administrators to access and query SECY MIB data from a router using SNMP. These examples assume the SNMP community string is set to public and the device management IP address is 10.0.0.1.

Obtaining the MACsec Controlled Port Interface Index

- To perform an SNMP walk on the entire SECY MIB:

```
snmpwalk -v2c -c public 10.0.0.1 1.0.8802.1.1.3
```

- To query the secyTxSCTable and obtain the TxSCI value for interface Gi0/1/0/0 (where the ifindex for MACsecControlled0/1/0/0 is 18):

```
snmpget -v2c -c public 10.0.0.1 iso.0.8802.1.1.3.1.1.2.1.1.18
```

These SNMP query examples help administrators retrieve security-related MIB data for monitoring and management of router interfaces.