



Configuring MACsec

This module describes how to configure Media Access Control Security (MACsec) encryption on Cisco 8000 Series Routers. MACsec is a Layer 2 IEEE 802.1AE standard for encrypting packets between two MACsec-capable routers.

- [Understanding MACsec Encryption, on page 1](#)
- [MKA Authentication Process, on page 2](#)
- [MACsec Frame Format, on page 3](#)
- [Advantages of Using MACsec Encryption, on page 3](#)
- [Hardware Support for MACsec, on page 4](#)
- [MACsec PSK, on page 4](#)
- [Fallback PSK, on page 5](#)
- [Configuring and Verifying MACsec Encryption , on page 6](#)
- [Creating a MACsec Keychain, on page 8](#)
- [Creating a User-Defined MACsec Policy, on page 12](#)
- [Applying MACsec Configuration on an Interface, on page 14](#)
- [Enable MACsec Mode on PHY, on page 20](#)
- [MACsec Policy Exceptions, on page 21](#)
- [Verifying MACsec Encryption on IOS XR, on page 22](#)
- [Verifying MACsec Encryption on Cisco 8000 Series Routers , on page 33](#)
- [MACsec SecY Statistics, on page 36](#)
- [Power-on Self-Test KAT for Common Criteria and FIPS, on page 44](#)
- [Related Commands for MACsec, on page 47](#)

Understanding MACsec Encryption

Security breaches can occur at any layer of the OSI model. At Layer 2, some of the common breaches are MAC address spoofing, ARP spoofing, Denial of Service (DoS) attacks against a DHCP server, and VLAN hopping.

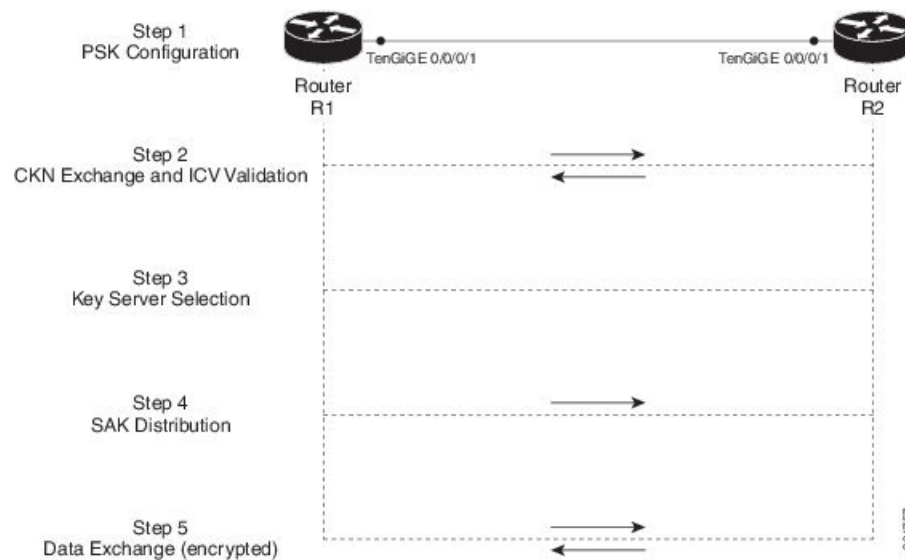
MACsec secures data on physical media, making it impossible for data to be compromised at higher layers. As a result, MACsec encryption takes priority over any other encryption method such as IPsec and SSL at higher layers. MACsec is configured on the Customer Edge (CE) router interfaces that connect to Provider Edge (PE) routers and on all the provider router interfaces.

MKA Authentication Process

MACsec provides the secure MAC Service on a frame-by-frame basis, using GCM-AES algorithm. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

The MACsec encryption process is illustrated in the following figure and description.

Figure 1: MKA Encryption Process



Step 1: When a link is first established between two routers, they become peers. Mutual peer authentication takes place by configuring a Pre-shared Key (PSK).

Step 2: On successful peer authentication, a connectivity association is formed between the peers, and a secure Connectivity Association Key Name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a Connectivity Association Key (CAK), which is effectively a secret key.

Step 3: A key server is selected between the routers, based on the configured key server priority. Lower the priority value, higher the preference for the router to become the key server. If no value is configured, the default value of 16 is taken to be the key server priority value for the router. Lowest priority value configures that router as the key server, while the other router functions as a key client. The following rules apply to key server selection:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each router selects a peer advertising the highest preference as its key server provided that peer has not selected another router as its key server or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the router with the highest priority SCI is chosen as key server (KS).

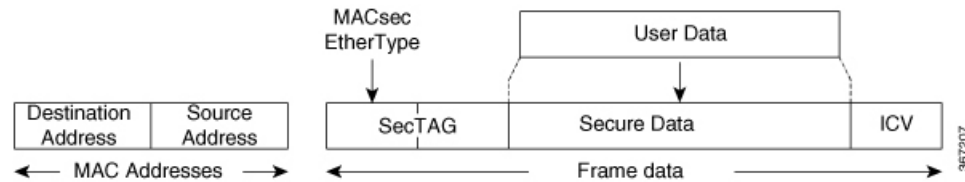
Step 4: A security association is formed between the peers. The key server generates and distributes the Secure Association Key (SAK) to the key client (peer). Each secure channel is supported by an overlapped sequence of Security Associations (SA). Each SA uses a new Secure Association Key (SAK).

Step 5: Encrypted data is exchanged between the peers.

MACsec Frame Format

The MACsec header in a frame consists of three components as illustrated in the following figure.

Figure 2: MACsec Frame Format



- **SecTAG:** The security tag is 8-16 bytes in length and identifies the SAK to be used for the frame. With Secure Channel Identifier (SCI) encoding, the security tag is 16 bytes in length, and without the encoding, 8 bytes in length (SCI encoding is optional). The security tag also provides replay protection when frames are received out of sequence.
- **Secure Data:** This is the data in the frame that is encrypted using MACsec and can be 2 or more octets in length.
- **ICV:** The ICV provides the integrity check for the frame and is usually 8-16 bytes in length, depending on the cipher suite. Frames that do not match the expected ICV are dropped at the port.

Advantages of Using MACsec Encryption

- **Data Integrity Check:** Integrity check value (ICV) is used to perform integrity check. The ICV is sent with the protected data unit and is recalculated and compared by the receiver to detect data modification.
- **Data Encryption:** Enables a port to encrypt outbound frames and decrypt MACsec-encrypted inbound frames.
- **Replay Protection:** When frames are transmitted through the network, there is a strong possibility of frames getting out of the ordered sequence. MACsec provides a configurable window that accepts a specified number of out-of-sequence frames.
- **Support for Clear Traffic:** If configured accordingly, data that is not encrypted is allowed to transit through the port.

Hardware Support for MACsec

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
MACsec capability for 88-LC0-36FH-M	Release 7.3.15	<p>The Cisco 8800 36x400GE QSFP56-DD Line Card with MACsec based on Q200 Silicon (88-LC0-36FH-M) supports point-to-point (P2P) MACsec capability with 400GE line rate encryption on its physical interfaces.</p> <p>This release also introduces point-to-multipoint (P2MP) MACsec capability on both 8800-LC-36FH-M and 8800-LC-48H (Cisco 8800 48x100 GbE QSFP28 Line Card based on Q100 Silicon). The P2MP topology connects multiple nodes to one central node, thereby avoiding unnecessary packet replication at the ingress router.</p>

The MACsec technology is supported on 48-port 100GE line cards and 36-port 400GE line cards.

Table 2: MACsec Hardware Support Matrix

Cisco IOS XR Software Release	PID	Description
Release 7.5.2	8202-32FH-M	Cisco 8202 32x400GE 2 Rack Unit (RU) Fixed Chassis
Release 7.3.15	88-LC0-36FH-M	Cisco 8800 36x400GE QSFP56-DD Line Card with MACsec based on Q200 Silicon
Release 7.0.12	8800-LC-48H	Cisco 8800 48x100 GbE QSFP28 Line Card based on Q100 Silicon

MACsec PSK

A pre-shared key includes a connectivity association key name (CKN) and a connectivity association key (CAK). A pre-shared key is exchanged between two devices at each end of a point-to-point (P2P) link to enable MACsec using static CAK security mode. The MACsec Key Agreement (MKA) protocol is enabled

after the pre-shared keys are successfully verified and exchanged. The pre-shared keys, the CKN and CAK, must match on both ends of a link.

For more information on MACsec PSK configuration, see [Applying MACsec Configuration on an Interface, on page 14](#).

Fallback PSK

Fallback is a session recovery mechanism when primary PSK fails to bring up secured MKA session. It ensures that a PSK is always available to perform MACsec encryption and decryption.

- In CAK rollover of primary keys, if latest active keys are mismatched, system performs a hitless rollover from current active key to fallback key, provided the fallback keys match.
- If a session is up with fallback, and primary latest active key configuration mismatches are rectified between peers, system performs a hitless rollover from fallback to primary latest active key.



Note

- A valid Fallback PSK (CKN and CAK) must be configured with infinite lifetime. If the fallback PSK is configured with CAK mismatch, the only recovery mechanism is to push a new set of PSK configurations (both on fallback PSK keychain and primary PSK chain, in that order) on all the association members.
- In P2P topologies, a rollover to the fallback PSK happens when either of the nodes in the Secure Association (SA) cannot peer up with the primary PSK. Whereas, in P2MP, the fallback happens only at the expiry or deletion of the primary key on all peers, not just on one of the peers. On deletion or expiry of the primary PSK on one of the nodes, say R1, a new key server is chosen among the peer nodes that does a SAK rekey for the remaining nodes. This ensures that R1 is no longer part of the SA, and the network drops all traffic to and from R1.

The following is a sample syslog for session secured with fallback PSK:

```
%L2-MKA-5-SESSION_SECURED_WITH_FALLBACK_PSK : (Hu0/1/0/0) MKA session secured, CKN:ABCD
```

For more information on MACsec fallback PSK configuration, see [Applying MACsec Configuration on an Interface, on page 14](#).

Active Fallback

The Cisco IOS XR Software Release 7.0.14 introduces the support for active fallback feature that initiates a fallback MKA session on having fallback configuration under the interface.

The key benefits of active fallback feature are:

- Faster session convergence on fallback, in the event of primary key deletion, expiry or mismatch.
- Faster traffic recovery under should-secure security policy when both primary and fallback mismatch happens.

With the introduction of active fallback functionality, the output of various MACsec show commands include the fallback PSK entry as well. If the session is secured with primary key, the fallback session will be in ACTIVE state. See, [Verifying MACsec Encryption on IOS XR, on page 22](#) for details and sample outputs.



Note If the peer device is running on an older release that does not support active fallback feature, you must configure the **enable-legacy-fallback** command under the **macsec-policy** to ensure backward compatibility.

Configuring and Verifying MACsec Encryption

MACsec can be configured on physical Ethernet interfaces or interface bundles (link bundles), as explained in this section.

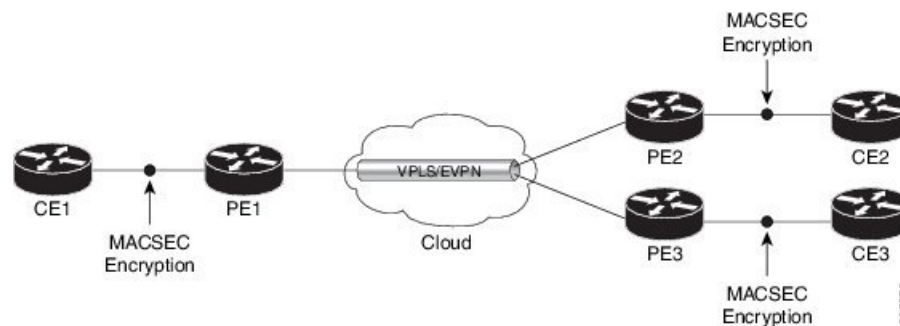


Note Enabling MACsec encryption on any on physical Ethernet interfaces or interface bundles (link bundles) will add an overhead of 32 bytes on the maximum transmission unit (MTU) size of link-state packets (LSPs). Therefore, you must set the LSP MTU to 32 bytes less than the interface MTU, to account for MACsec overhead. Use the **lsp-mtu** command to configure the maximum transmission unit (MTU) size of link-state packets (LSPs) on each router where MACsec is enabled.

Use Case 1: MACsec in a VPLS/EVPN

A typical VPLS network often suffers the injection of labeled traffic from potential hackers. The following figure illustrates the use of MACsec in a VPLS/EVPN network for encrypting the data being exchanged over the VPLS cloud. In this topology MACsec is configured on the PE-facing interfaces of the CE routers.

Figure 3: MACsec in a VPLS/EVPN Cloud



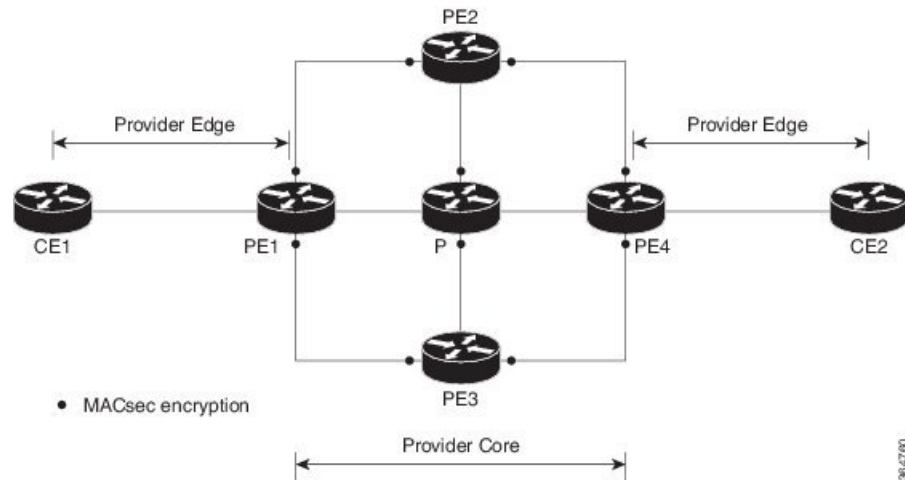
Use Case 2: MACsec in an MPLS Core Network

MACsec in an MPLS core network can be configured on physical interfaces or link bundles (Link Aggregation Group or LAG).

In the following topology, MACsec is configured on all router links in the MPLS core. This deployment is useful when the MPLS network spans data centers that are not co-located in the same geography. Each link is, therefore, a link between two data centers and all data exchanged is encrypted using MACsec.

The following figure illustrates the use of MACsec on physical interfaces in an MPLS core network.

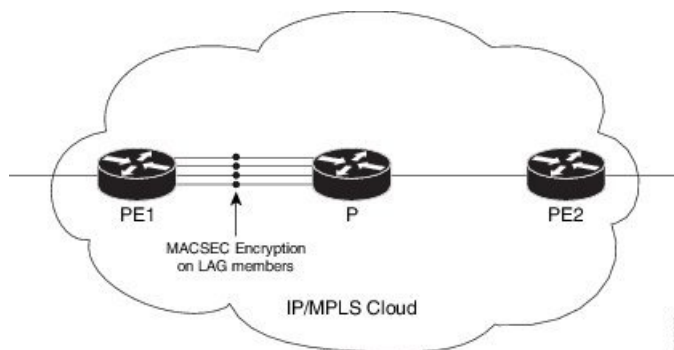
Figure 4: MACsec on Physical Interfaces in an MPLS Core Network



When MACsec is configured on the members of a LAG, an MKA session is set up for each member. SAK is exchanged for each LAG member and encryption or decryption takes place independently of other members in the group.

The following figure illustrates the use of MACsec on a link bundle in an MPLS core network.

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



The following section describes procedures for configuring and verifying MACsec configuration in the described deployment modes.

Prior to configuring MACsec on a router interface, the MACsec keychain must be defined. If you apply the MACsec keychain on the router without specifying a MACsec policy, the default policy is applied. A default MACsec policy is pre-configured with default values. If you need to change any of the pre-configured values, create a different MACsec policy.

Configuring MACsec involves the following steps:

1. Creating a MACsec keychain
2. Creating a user-defined MACsec policy
3. Applying MACsec configuration on physical interfaces

Creating a MACsec Keychain

A MACsec keychain is a collection of keys used to authenticate peers needing to exchange encrypted information. While creating a keychain, we define the key(s), key string with password, the cryptographic algorithm, and the key lifetime.

MACsec Keychain Keyword	Description
Key	The MACsec key or the CKN can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.
Key-string	The MACsec key-string or the CAK can be either 32 characters or 64 characters in length (32 for AES-128, 64 for AES-256).
Lifetime	This field specifies the validity period of a key. It includes a start time, and an expiry time. We recommend you to set the value for expiry time as <i>infinite</i> .

Guidelines for Configuring MACsec Keychain

MACsec keychain management has the following configuration guidelines:

- To establish MKA session, ensure that the MACsec key (CKN) and key-string (CAK) match at both ends.
- MKA protocol uses the latest active key available in the Keychain. This key has the latest Start Time from the existing set of currently active keys. You can verify the values using the **show key chain keychain-name** command.
- Deletion or expiry of current active key brings down the MKA session resulting in traffic hit. We recommend you to configure the keys with infinite lifetime. If fallback is configured, traffic is safeguarded using fallback on expiry or deletion of primary-keychain active key.
- To achieve successful key rollover (CAK-rollover), the new key should be configured such that it is the latest active key, and kicks-in before the current key expires.
- We recommend an overlap of at least one minute for hitless CAK rollover from current key to new key.
- Start time and Expiry time can be configured with future time stamps, which allows bulk configuration for daily CAK rotation without any intervention of management agent.
- From Cisco IOS XR Software Release 7.2.1 and later, the MACsec key IDs (configured through CLI using the **macsec key** command under the key chain configuration mode) are considered to be case insensitive. These key IDs are stored as uppercase letters. For example, a key ID of value 'FF' and of value 'ff' are considered to be the same, and both these key IDs are now stored in uppercase as 'FF'. Whereas, prior to Release 7.2.1, both these values were treated as case sensitive, and hence considered as two separate key IDs. Hence it is recommended to have unique strings as key IDs for a MACsec key chain to avoid flapping of MACsec sessions. However, the support for this case insensitive IDs is applicable only for the configurations done through CLI, and not for configurations done through Netconf protocol.

Also, it is recommended to do a prior check of the MACsec key IDs before upgrading to Release 7.2.1 or later.

Consider a scenario where two MACsec key IDs with the same set of characters (say, ff and FF) are configured under the same key chain.

```
key chain 1
 macsec
  key ff
    lifetime 02:01:01 may 18 2020 infinite
  !
  key FF
    lifetime 01:01:01 may 18 2020 infinite
```

When you upgrade to Release 7.2.1 or later, only one of these key IDs is retained. That is 'FF', the one that was applied second in this example.

Follow these steps to configure a MACsec keychain:

Keychain Name: Provide a name for the MACsec keychain.

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

MACsec Mode: Enter the MACsec mode.

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

MACsec key: Provide a name for the MACsec key.

The MACsec key can be up to 64 characters in length. The key must be of an even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.

You can also configure a fall-back pre-shared key(PSK) to ensure that a PSK is always available to perform MACsec encryption and decryption. The fallback PSK along with the primary PSK ensures that the session remains active even if the primary PSK is mismatched or there is no active key for the primary PSK.

The configured key is the CKN that is exchanged between the peers.

See the guidelines section to know more about the need for a unique key ID for a MACsec key chain.



Note If you are configuring MACsec to interoperate with a MACsec server that is running software prior to Cisco IOS XR Release 6.1.3, then ensure that the MACsec key length is of 64 characters. You can add extra zero characters to the MACsec key so that the length of 64-characters is achieved. If the key length is lesser than 64 characters, authentication will fail.

```
Router(config-kc-MacSec)#key 1234
```

AES Encryption: Enter the key string and the cryptographic algorithm to be used for the key.

```
! AES 128-bit encryption

Router(config-kc-MacSec-1234)#key-string 12345678
Router(config-kc-MacSec-1234)#cryptographic-algorithm AES-128-CMAC-96
```

MACsec key (CKN) lifetime or validity period: The lifetime period can be configured as a validity period between two dates (for example, Jan 01 2019 to Dec 31 2019), or with infinite validity. The key is valid from the time you configure (in HH:MM:SS format). Duration is configured in seconds.



Note When a key has expired, the MACsec session is torn down and running the **show macsec mka session** command does not display any information. If you run the **show macsec mka interface detail** command, the output displays ***** No Active Keys Present ***** in the PSK information.

```
Router(config-kc-MacSec-1234)#lifetime 05:00:00 01 January 2019 duration 1800
! Configuring lifetime for a defined period

Router(config-kc-MacSec-1234)#lifetime 05:00:00 20 february 2019 12:00:00 30 september 2019

! Configuring lifetime as infinite

Router(config-kc-MacSec-1234)#lifetime 05:00:00 01 January 2019 infinite
Router(config-kc-MacSec-1234)#commit
```

Securing the MACsec Pre-shared Key (PSK) Using Type 6 Password Encryption

Using the Type 6 password encryption feature, you can securely store MACsec plain text key string (CAK) in Type 6 encrypted format.

The primary key is the password or key used to encrypt all plain text MACsec key strings (CAK) in the router configuration with the use of an Advance Encryption Standard (AES) symmetric cipher. The primary key is not stored in the router configuration and cannot be seen or obtained in any way while connected to the router.

The Type 6 password encryption is effective only if a primary key is configured. The Type 6 Password Encryption is currently available on Cisco 8000 Series Routers.

Configuring a Primary Key and Enabling the Type 6 Password Encryption Feature

You can configure a primary key for Type 6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature for securing the MACsec keys (key string/CAK). When prompted, enter the primary key details. The primary key can contain between 6 and 64 alphanumeric characters.

Primary Key Creation

```
Router#key config-key password-encryption
New password Requirements: Min-length 6, Max-length 64
Characters restricted to [A-Z][a-z][0-9]
Enter new key :
Enter confirm key :
```

Type 6 password encryption

```
Router#configure terminal
Router(config)#password6 encryption aes
Router(config)#commit
```

- **Modifying the Primary Key** - If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

Modifying a primary key would re-encrypt all the existing Type 6 format key strings with the new primary key. If Type 6 key strings are present, ensure that the **password6 configuration aes** command is present to enable re-encryption with the new primary key. Otherwise, the primary key update operation fails.

- **Deleting the Primary Key** - Follow these steps to delete the primary key at any time.

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



Note 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.

Configuring MACsec Pre-shared Key (PSK) For Type 6 Password Encryption

Ensure that you have configured a primary key using the **key config-key password-encryption** command and enabled the Type 6 encryption feature using the **password6 encryption aes** command.

```
Router#configure terminal
Router(config)#key chain kcl macsec
Router(config-kcl-MacSec)# key 1111
```

! Configuring 32 byte hex CAK

```
Router(config-kcl-MacSec-1111)# key-string 12345678901234567890123456789022
cryptographic-algorithm aes-128-cmac
```

! Configuring 64 byte hex CAK

```
Router(config-kcl-MacSec-1111)# key-string
1234567890123456789012345678902212345678901234567890123456789022 cryptographic-algorithm
aes-256-cmac
Router(config-kcl-MacSec-1111)# lifetime 00:00:00 1 October 2019 infinite
Router(config-kcl-MacSec-1111)# commit
```

Running Configuration

The following is a sample output of the **show running-config key chain kcl** command. The MACsec key chain name, the Type 6 key, and key-string information are displayed.

```
Router#show running-config key chain kcl

key chain kcl
 macsec
  key 1111
    key-string password6 674c434d695b5c5b464f546854474d58504946535455644b5043685969454861685
645464c5047635c464b5a6847566751535f45475266635256645a4a49454b4d646751454543465254684957665f6149674c5e4d
```

```
516843484446575c49685f5648665a604b5450435952634c61455d4b5e414142 cryptographic-algorithm
aes-256-cmac
    lifetime 00:00:00 october 01 2019 infinite
!
```

Creating a User-Defined MACsec Policy

- **MACsec policy** - Create a MACsec policy and configure the cipher suite to be used for MACsec encryption, including the confidentiality offset.



Note We recommend to change the offset value of the **conf-offset** *<offset_value>* command (MACsec encryption command) in the routers only when the port is in **admin down** state (that is, when the interface is shut down). Changing the offset value otherwise may result in traffic loss.

In this example, the GCM-AES-XPB-256 encryption algorithm is used. A 256-bit encryption algorithm uses a larger key that requires more rounds of hacking to be cracked. 256-bit algorithms provide better security against large mass security attacks, and include the security provided by 128-bit algorithms. Extended Packet Numbering (XPN) is used to reduce the number of key rollovers while data is sent over high speed links. It is therefore highly recommended to use GCM-AES-XPB-256 encryption algorithm for higher data ports.



Note For Cisco 8000 Series Routers to interoperate with Cisco ASR9000 Series Routers that are older than Release 6.2.3, configure a user defined MACsec policy with the `policy-exception lacp-in-clear` command to bring up the MKA sessions over bundle interfaces running in LACP modes.

```
Router# configure terminal
Router(config)# macsec-policy mp-SF
Router(config-macsec-policy)# cipher-suite GCM-AES-XPB-128
Router(config-macsec-policy)# conf-offset CONF-OFFSET-30
```

- **Key server priority** - You can enter a value between 0-255. Lower the value, higher the preference to be selected as the key server. In this example, a value of 0 configures the router as the key server, while the other router functions as a key client. The key server generates and maintains the SAK between the two routers. The default key server priority value is 16.

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

- **Security policy parameters** - Enable the Must-Secure or Should-Secure parameter.

Must-Secure imposes that only MACsec encrypted traffic can flow. Hence, until the MKA session is secured, traffic is dropped.

Should-Secure allows unencrypted traffic to flow until the MKA session is secured. After the MKA session is secured, only encrypted traffic can flow.



Note Unlike in P2P scenarios, the traffic drops in P2MP if a third peer that is added with **should-secure** security policy fails to establish secure connection with two other peers that have already established a secure connection using **should-secure**. This is because, the first two peers update their configurations to drop all untagged (unencrypted) packets, once they establish a secure connection.

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

- **Data delay protection** - Data delay protection allows MKA participants to ensure that the data frames protected by MACsec are not delayed by more than 2 seconds. Each SecY uses MKA to communicate the lowest PN used for transmission with the SAK within two seconds. Traffic delayed longer than 2 seconds are rejected by the interfaces enabled with delay protection.

By default, the data delay protection feature is disabled. Configuring the `delay-protection` command under MACsec-policy attached to MACsec interface will enable the data delay protection feature on that interface. When enabled, the `show macsec mka session interface interface detail` command output displays the **Delay Protection** field as **TRUE**.

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

- **Replay protection window size** - The window size dictates the maximum out-of-sequence frames that are accepted. You can configure a value between 0 and 1024.
- **ICV for the frame arriving on the port** - This parameter configures inclusion of the optional ICV Indicator as part of the transmitted MACsec Key Agreement PDU (MKPDU). This configuration is necessary for MACsec to interoperate with routers that run software prior to IOS XR version 6.1.3. This configuration is also important in a service provider WAN setup where MACsec interoperates with other vendor MACsec implementations that expect ICV indicator to be present in the MKPDU.

```
Router(config-macsec-policy)# window-size 64
Router(config-macsec-policy)# include-icv-indicator
Router(config-macsec-policy)# commit
```

Running Configuration

The following is a sample output of the `show running-config macsec-policy` command.

```
Router# show running-config macsec-policy mp-SF

macsec-policy mp-SF
  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
!
```

MACsec SAK Rekey Interval

You can set a timer value to rekey the MACsec secure association key (SAK) at a specified interval. This periodic refresh of SAK ensures that data encryption key is frequently updated. The configuration is effective on the node acting as a key server.

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.

Configuration Example

```
Router#configure
Router(config)#macsec-policy test-policy
Router(config-macsec-policy)#sak-rekey-interval 120
Router(config-macsec-policy)#commit
```

Running Configuration

```
macsec-policy test-policy
  sak-rekey-interval 120
!
```

Associated Command

sak-rekey-interval

Applying MACsec Configuration on an Interface

The MACsec service configuration is applied to the host-facing interface of a CE router.

Guidelines for MACsec Interface Configuration

- Configure different keychains for primary and fallback PSKs.
- We do not recommend to update both primary and fallback PSKs simultaneously, because fallback PSK is intended to recover MACsec session on primary key mismatch.
- When using MACsec, we recommend you adjust the maximum transmission unit (MTU) of an interface to accommodate the MACsec overhead. Configuring MTU value on an interface allows protocols to do MTU negotiation including MACsec overhead. For instance, if the default MTU is 1514 bytes, configure the MTU to 1546 bytes (1514 + 32).
- The minimum MTU for IS-IS protocol on the MACsec interface is 1546 bytes.
- For enabling MACsec on bundle members :
 - We recommend configuring the maximum possible MTU on the bundle interface.
 - The MTU configurations must account for the maximum packet size of the protocols running on the bundle interface and 32 bytes of MACsec overhead.
 - For IS-IS protocol running on the bundle interface, hello-padding must be disabled.



Tip You can programmatically view the MACsec configuration using the `openconfig-macsec.yang` OpenConfig data model. To get started with using data models, see *Programmability Configuration Guide for Cisco 8000 Series Routers*.

MACsec PSK Configuration on an Interface

```
Router#configure
Router(config)#interface hundredGigE 0/1/0/10
Router(config-if)# macsec psk-keychain kc policy mp-SF
```

To enable MACsec PSK on a physical interface without the MACsec policy, use this command:

```
Router(config-if)#macsec psk-keychain script_kc
```

MACsec Fallback PSK Configuration on an Interface

It is optional to configure a fallback PSK. If a fallback PSK is configured, the fallback PSK along with the primary PSK ensures that the session remains active even if the primary PSK is mismatched, or there is no active key for the primary PSK.

```
Router(config-if)#macsec psk-keychain kc fallback-psk-keychain fallback_kc policy mp-SF
Router(config-if)#commit
```

Verification

Before the introduction of active fallback functionality:

```
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                    : kc
Member Identifier (MI)      : C12A70FEE1212B835BDDDCBA
Message Number (MN)        : 395
Authenticator               : NO
Key Server                  : NO
MKA Cipher Suite            : 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         : 16
Delay Protection            : FALSE
Replay Window Size         : 64
Include ICV Indicator       : FALSE
Confidentiality Offset      : 30
Algorithm Agility           : 80C201
```

```
SAK Cipher Suite           : 0080C20001000003 (GCM-AES-XPB-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:

```
-----
          MI              MN              Rx-SCI              SSCI  KS-Priority
-----
018E2F0D63FF2ED6A5BF270E      86      008a.962d.7400/0001      2      16
```

Potential Peer List:

```
-----
          MI              MN              Rx-SCI              SSCI  KS-Priority
-----
```

Peers Status:

```
Last Tx MKPDU           : 2019 Oct 08 07:39:56.905
Peer Count              : 1
```

```
RxSCI                   : 008A962D74000001
MI                       : 018E2F0D63FF2ED6A5BF270E
Peer CAK                 : Match
Latest Rx MKPDU         : 2019 Oct 08 07:39:57.363
```

With the introduction of active fallback functionality:

The following is a snippet from the sample output that displays entry for active fallback PSK as well. The secured primary session output part is truncated here, which is exactly same as the output given above.

```
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             : 0257.3fae.5cda/0001
Local Tx-SSCI           : 1
```

```
- - - - -
- - - - -
- - - - -
```

MKA Detailed Status for MKA Session

```
=====
```

```
Status: Active - Marked Peer as Live (Waiting for SAK generation/distribution)
```

```
Local Tx-SCI             : 0257.3fae.5cda/0001
Local Tx-SSCI           : 1
Interface MAC Address    : 0257.3fae.5cda
MKA Port Identifier      : 1
Interface Name           : Hu0/1/0/0
CAK Name (CKN)          : 1111
CA Authentication Mode   : FALLBACK-PSK
Keychain                 : fb1
Member Identifier (MI)   : FC53A31E030E385981E0AACE
Message Number (MN)     : 178
Authenticator           : NO
Key Server               : NO
MKA Cipher Suite         : AES-256-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPB-256
Key Distribution Mode    : SAK
```



```

Latest SAK Status           : Rx & Tx
Latest SAK AN              : 1
Latest SAK KI (KN)        : 725FF8F6605A3D428972538F00000001 (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
Time to exit suspension   : 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 : 0
    
```

```

Live Peer List:
-----
          MI              MN              Rx-SCI              SSCI  KS-Priority
-----
6A894FE1E984AD5314F33D21      188      0201.9ab0.85af/0001      0      16
    
```

```

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

```

Peers Status:
Last Tx MKPDU           : 2021 Apr 30 14:56:33.797
Peer Count              : 1

RxSCI                   : 02019AB085AF0001
MI                      : 6A894FE1E984AD5314F33D21
Peer CAK                 : Match
Latest Rx MKPDU         : 2021 Apr 30 14:56:34.638
    
```

The following is a sample output of the **show macsec mka session interface** command. With this command, you can verify whether the interface of the router is peering with its neighbor after MACsec configuration. The MACsec PSK validation detects inconsistency or mismatch of primary and fallback keys (CAK) being used by MKA, allowing operators to rectify the mismatch.

Verify whether the MKA session is secured with MACsec on the respective interface. The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

```
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

```

Before the introduction of active fallback functionality:

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

```

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                : kc
Member Identifier (MI)  : C12A70FEE1212B835BDDDCBA
Message Number (MN)    : 433
Authenticator          : NO
Key Server              : NO
MKA Cipher Suite       : 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     : 16
Delay Protection        : FALSE
Replay Window Size     : 64
Include ICV Indicator  : FALSE
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:

```

-----
MI              MN              Rx-SCI          SSCI  KS-Priority
-----
018E2F0D63FF2ED6A5BF270E  123      008a.962d.7400/0001  2      16

```

Potential Peer List:

```

-----
MI              MN              Rx-SCI          SSCI  KS-Priority
-----

```

```

Peers Status:
  Last Tx MKPDU      : 2019 Oct 08 07:41:12.929
  Peer Count         : 1

  RxSCI              : 008A962D74000001
  MI                  : 018E2F0D63FF2ED6A5BF270E
  Peer CAK            : Match
  Latest Rx MKPDU    : 2019 Oct 08 07:41:11.400

```

With the introduction of active fallback functionality:

The following is a snippet from the sample output that displays entry for active fallback PSK as well. The secured primary session output part is truncated here, which is exactly same as the output given above.

```

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

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

Local Tx-SCI          : 7872.5d1a.e7d4/0001
Local Tx-SSCI        : 1
- - - - -
- - - - -
- - - - -
MKA Detailed Status for MKA Session
=====
Status: Active - Marked Peer as Live (Waiting for SAK generation/distribution)

Local Tx-SCI          : 34ed.1b5b.d0d7/0001
Local Tx-SSCI        : 1
Interface MAC Address : 34ed.1b5b.d0d7
MKA Port Identifier   : 1
Interface Name        : Hu0/4/0/27
CAK Name (CKN)       : 2222
CA Authentication Mode : FALLBACK-PSK
Keychain              : fb1
Member Identifier (MI) : C0978A6B0916C3FC959773FE
Message Number (MN)   : 24039
Authenticator         : NO
Key Server            : NO
MKA Cipher Suite      : AES-256-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPB-256

Latest SAK Status     : Rx & Tx
Latest SAK AN         : 2
Latest SAK KI (KN)   : 3D008A7D75DF0A9A35F9E3A900000002 (2)
Old SAK Status        : No Rx, No Tx
Old SAK AN            : 1
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
Time to exit suspension : NA

MKA Policy Name       : r1
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-XPB-256)

```

```

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:
-----
          MI                MN                Rx-SCI                SSCI  KS-Priority
-----
B5ED6849883F34FEE89F74D1      26068          008a.9681.c02c/0001      2      16

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

Peers Status:
Last Tx MKPDU              : 2021 Apr 28 02:08:03.795
Peer Count                  : 1

RxSCI                      : 008A9681C02C0001
MI                          : B5ED6849883F34FEE89F74D1
Peer CAK                    : Match
Latest Rx MKPDU            : 2021 Apr 28 02:08:02.749

```

Enable MACsec Mode on PHY

You can enable the MACsec mode for the physical layer transceiver (PHY) of a line card by using the **hw-module macsec-mode** command in XR Config mode mode.

Configuration Example

```

Router#configure
Router(config)#hw-module macsec-mode location 0/1/CPU0
Router(config)#commit

```

You must reload the line card for this configuration to take effect.



Note If the MACsec mode is already enabled on a node such as a line card, then the system does not allow you to configure the **hw-module macsec-mode location all** command again. This restriction is in place to prevent conflicts in configuration, especially in a configuration restore scenario. In such scenarios, you can make use of the **show hw-module macsec-mode** command to know of the respective running configurations in place.

Running Configuration

```

hw-module macsec-mode location 0/1/CPU0
!

```

Verification

You can use the show **hw-module macsec-mode** command in the XR EXEC mode to display the MACsec mode of line cards, and the user action to be performed.

```
Router#show hw-module macsec-mode location 0/1/CPU0
Sat Dec 7 14:31:52.668 UTC
Location          Configured      Running         Action
-----
0/1/CPU0          YES             NO              RELOAD
```

You can also use the **show hw-module macsec-mode location all** command to display the MACsec mode information of all nodes. This **location all** option is available starting Cisco IOS XR Software Release 7.0.14.

```
Router#show hw-module macsec-mode location all
Sun Feb 16 21:06:07.726 UTC
Location          Configured      Running         Action
-----
0/0/CPU0          YES             NO              RELOAD
0/7/CPU0          NO              NO              NONE
```

MACsec Policy Exceptions

By default, the MACsec security policy uses **must-secure** option, that mandates data encryption. Hence, the packets cannot be sent in clear-text format. To optionally bypass the MACsec encryption or decryption for Link Aggregation Control Protocol (LACP) packets, and to send the packets in clear-text format, use the **policy-exception lacp-in-clear** command in macsec-policy configuration mode. This functionality is beneficial in scenarios such as, in a network topology with three nodes, where bundles are terminated at the middle node, whereas MACsec is terminated at the end nodes.

This MACsec policy exception is also beneficial in interoperability scenarios where the node at the other end expects the data packets to be in clear text.

From Cisco IOS XR Software Release 7.3.1 and later, an alternative option, **allow**, is introduced under the macsec-policy configuration mode, that allows packets to be sent in clear-text format. You can use the **allow lacp-in-clear** command for LACP packets.

Similarly, you can use the **allow pause-frames-in-clear** to allow Ethernet PAUSE frame packets in clear text, from Release 7.3.15 and later.

How to Create MACsec Policy Exception



Note The **policy-exception lacp-in-clear** command under macsec-policy configuration mode is deprecated. Hence, it is recommended to use the **allow lacp-in-clear** command instead, to allow LACP packets in clear-text format.

Configuration Example

Using the **policy-exception** command:

```
Router#configure
Router(config)#macsec-policy test-macsec-policy
Router(config-macsec-policy)#policy-exception lacp-in-clear
Router(config-macsec-policy)#commit
```

Using the **allow** command:

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

Running Configuration

With the **policy-exception** command:

```
Router#show run macsec-policy test-macsec-policy
macsec-policy test-macsec-policy
  policy-exception lacp-in-clear
  security-policy should-secure
  include-icv-indicator
  sak-rekey-interval seconds 120
!
```

With the **allow** command:

```
Router#show run macsec-policy test-macsec-policy
macsec-policy test-macsec-policy
  allow lacp-in-clear
  allow pause-frames-in-clear
  security-policy should-secure
  include-icv-indicator
  sak-rekey-interval seconds 120
!
```

Associated Commands

- **policy-exception lacp-in-clear**
- **allow lacp-in-clear**
- **allow pause-frames-in-clear**

Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode.



Note With the introduction of active fallback functionality in Cisco IOS XR Software Release 7.0.14, the output of various MACsec show commands include the fallback PSK entry as well.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec policy configuration.
2. Verify the MACsec configuration on the respective interface.
3. Verify whether the interface of the router is peering with its neighbor after MACsec configuration.
4. Verify whether the MKA session is secured with MACsec on the respective interface.
5. Verify the MACsec session counter statistics.

DETAILED STEPS

Step 1 Verify the MACsec policy configuration.

Example:

```
Router# show macsec policy mp-SF
```

```
=====
Policy          Cipher          Key-Svr   Window   Conf      Delay
name            Suite           Priority  Size    Offset    Protection
=====
mp-SF           GCM-AES-128    16        64      30        FALSE
=====
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

Step 2 Verify the MACsec configuration on the respective interface.

You can verify the MACsec encryption on the configured interface bundle (MPLS network) or P2MP interface (VPLS network).

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-128  kc        PRIMARY  1234

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

```

With the introduction of active fallback functionality:

The following is a sample output that displays active fallback PSK entry as well:

```
Router# show macsec mka summary
Wed Apr 28 01:50:57.543 UTC
```

```
NODE: node0_4_CPU0
```

```
=====
Interface-Name      Status      Cipher-Suite      KeyChain      PSK/EAP      CKN
=====
Hu0/4/0/27          Secured    GCM-AES-XPN-256   kc1           PRIMARY      1111
Hu0/4/0/27          Active     GCM-AES-XPN-256   fb1           FALLBACK    2222
```

```
NODE: node0_6_CPU0
```

```
=====
Interface-Name      Status      Cipher-Suite      KeyChain      PSK/EAP      CKN
=====
Hu0/6/0/29          Secured    GCM-AES-XPN-256   kc1           PRIMARY      1111
Hu0/6/0/29          Active     GCM-AES-XPN-256   fb1           FALLBACK    2222
```

```
Total MACSec Sessions : 4
Secured Sessions       : 2
Pending Sessions       : 0
Suspended Sessions     : 0
Active Sessions        : 2
```

Run the **show run macsec-policy** command in the privileged executive mode to troubleshoot the configuration entered.

Step 3

Verify whether the interface of the router is peering with its neighbor after MACsec configuration.

The MACsec PSK validation detects inconsistency or mismatch of primary and fallback keys (CAK) being used by MKA, allowing operators to rectify the mismatch. The **#Peers** field in the output confirms the number of peers you have configured on the physical interface. If the number of peers is not reflected accurately in this output, run the **show run** command and verify the peer configuration on the interface.

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
```

The following is a sample output that displays active fallback PSK entry as well:

```
Router# show macsec mka session
Wed Apr 28 01:59:39.478 UTC
```

```
NODE: node0_4_CPU0
```

```
=====
Interface-Name      Local-TxSCI      #Peers  Status  Key-Server  PSK/EAP  CKN
=====
Hu0/4/0/27          34ed.1b5b.d0d7/0001  1       Secured  NO          PRIMARY  1111
Hu0/4/0/27          34ed.1b5b.d0d7/0001  1       Active  NO          FALLBACK  2222
```

```
NODE: node0_6_CPU0
```

```
=====
Interface-Name      Local-TxSCI      #Peers  Status  Key-Server  PSK/EAP  CKN
=====
Hu0/6/0/29          00d6.fee7.fe44/0001  1       Secured  YES         PRIMARY  1111
Hu0/6/0/29          00d6.fee7.fe44/0001  1       Active  YES         FALLBACK  2222
```


Note In the VPLS network, because of the configuration on a multipoint interface, the number of live peers displayed is more than 1.

```
Router#show macsec mka session
Mon Nov 23 11:20:39.835 UTC

NODE: node0_4_CPU0
=====
Interface-Name Local-TxSCI #Peers Status Key-Server PSK/EAP CKN
=====
FH0/4/0/34 34ed.1b5b.d10f/0001 2 Secured YES PRIMARY 1111
FH0/4/0/34 34ed.1b5b.d10f/0001 2 Active YES FALLBACK 2222
Hu0/4/0/28 34ed.1b5b.d0df/0001 2 Secured NO PRIMARY 1111
Hu0/4/0/28 34ed.1b5b.d0df/0001 2 Active NO FALLBACK 2222
Hu0/4/0/29 34ed.1b5b.d0e7/0001 2 Secured NO PRIMARY 1111
Hu0/4/0/29 34ed.1b5b.d0e7/0001 2 Active NO FALLBACK 2222
```

Before the introduction of active fallback functionality:

The following show command output verifies if the primary and fallback keys (CAK) are mismatched on both peer ends.

```
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                    : kc
Member Identifier (MI)     : C12A70FEE1212B835BDDDCBA
Message Number (MN)       : 3009
Authenticator              : NO
Key Server                  : NO
MKA Cipher Suite           : AES-128-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPB-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       : 16
Delay Protection          : FALSE
Replay Window Size       : 64
Include ICV Indicator     : FALSE
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:

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

Potential Peer List:

```
-----
                MI                MN                Rx-SCI                SSCI  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
```

With the introduction of active fallback functionality:

The following is a snippet from the sample output that displays entry for active fallback PSK as well. The secured primary session output part is truncated here, which is exactly same as the output given above.

```
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           : 0257.3fae.5cda/0001
Local Tx-SSCI         : 1
- - - - -
- - - - -
- - - - -
```

MKA Detailed Status for MKA Session

```
=====
```

Status: Active - Marked Peer as Live (Waiting for SAK generation/distribution)

```
Local Tx-SCI           : 0257.3fae.5cda/0001
Local Tx-SSCI         : 1
Interface MAC Address  : 0257.3fae.5cda
MKA Port Identifier    : 1
Interface Name        : Hu0/1/0/0
CAK Name (CKN)        : 1111
CA Authentication Mode : FALLBACK-PSK
Keychain              : fb1
Member Identifier (MI) : FC53A31E030E385981E0AACE
Message Number (MN)   : 178
Authenticator         : NO
Key Server            : NO
MKA Cipher Suite      : AES-256-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPN-256
Key Distribution Mode  : SAK
```

```

Latest SAK Status           : Rx & Tx
Latest SAK AN              : 1
Latest SAK KI (KN)        : 725FF8F6605A3D428972538F00000001 (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
Time to exit suspension   : 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-XPB-256)
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:

MI	MN	Rx-SCI	SSCI	KS-Priority
6A894FE1E984AD5314F33D21	188	0201.9ab0.85af/0001	0	16

Potential Peer List:

MI	MN	Rx-SCI	SSCI	KS-Priority
----	----	--------	------	-------------

Peers Status:

```

Last Tx MKPDU           : 2021 Apr 30 14:56:33.797
Peer Count              : 1

RxSCI                  : 02019AB085AF0001
MI                    : 6A894FE1E984AD5314F33D21
Peer CAK               : Match
Latest Rx MKPDU       : 2021 Apr 30 14:56:34.638
    
```

Note If the MKA session status is shown as **Secured** with **0 (Zero)** peer count, this means that the link is locally secured (Tx). This is because of MKA peer loss caused by **No Rx Packets (MKA Packet)** from that peer.

Step 4 Verify whether the MKA session is secured with MACsec on the respective interface.

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
=====
    
```

Before the introduction of active fallback functionality:

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

```
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                : kc
Member Identifier (MI)  : C12A70FEE1212B835BDDDCBA
Message Number (MN)    : 3058
Authenticator          : NO
Key Server              : NO
MKA Cipher Suite       : AES-128-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPB-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     : 16
Delay Protection        : FALSE
Replay Window Size     : 64
Include ICV Indicator   : FALSE
Confidentiality Offset  : 30
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000003 (GCM-AES-XPB-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:
```

```
-----
MI              MN              Rx-SCI          SSCI  KS-Priority
-----
018E2F0D63FF2ED6A5BF270E  2748  008a.962d.7400/0001  2      16
```

```
Potential Peer List:
```

```
-----
MI              MN              Rx-SCI          SSCI  KS-Priority
-----
```

```
Peers Status:
```

```
Last Tx MKPDU      : 2019 Oct 08 09:08:44.506
Peer Count         : 1
```

```
RxSCI              : 008A962D74000001
MI                  : 018E2F0D63FF2ED6A5BF270E
```

```
Peer CAK           : Match
Latest Rx MKPDU   : 2019 Oct 08 09:08:44.081
```

The **Status** field in the output confirms that the respective interface is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **INITIALIZING**.

With the introduction of active fallback functionality:

The following is a snippet from the sample output that displays entry for active fallback PSK as well. The secured primary session output part is truncated here, which is exactly same as the output given above.

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

```
MKA Detailed Status for MKA Session
```

```
=====
```

```
Status: Secured - Secured MKA Session with MACsec
```

```
Local Tx-SCI           : 7872.5d1a.e7d4/0001
Local Tx-SSCI          : 1
```

```
- - - - -
```

```
- - - - -
```

```
- - - - -
```

```
MKA Detailed Status for MKA Session
```

```
=====
```

```
Status: Active - Marked Peer as Live (Waiting for SAK generation/distribution)
```

```
Local Tx-SCI           : 34ed.1b5b.d0d7/0001
Local Tx-SSCI          : 1
Interface MAC Address   : 34ed.1b5b.d0d7
MKA Port Identifier     : 1
Interface Name          : Hu0/4/0/27
CAK Name (CKN)         : 2222
CA Authentication Mode  : FALLBACK-PSK
Keychain                : fb1
Member Identifier (MI)  : C0978A6B0916C3FC959773FE
Message Number (MN)    : 24039
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-256-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPB-256
```

```
Latest SAK Status      : Rx & Tx
Latest SAK AN          : 2
Latest SAK KI (KN)    : 3D008A7D75DF0A9A35F9E3A90000002 (2)
Old SAK Status         : No Rx, No Tx
Old SAK AN             : 1
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
Time to exit suspension : NA
```

```
MKA Policy Name        : r1
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-XPB-256)
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:
-----
                MI                MN                Rx-SCI                SSCI  KS-Priority
-----
B5ED6849883F34FEE89F74D1    26068    008a.9681.c02c/0001    2        16

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

Peers Status:
Last Tx MKPDU      : 2021 Apr 28 02:08:03.795
Peer Count        : 1

RxSCI              : 008A9681C02C0001
  MI               : B5ED6849883F34FEE89F74D1
  Peer CAK        : Match
  Latest Rx MKPDU : 2021 Apr 28 02:08:02.749

```

In a VPLS network with multipoint interface, the output would display more than one peer as follows:

```

Router#show macsec mka session interface Hu0/3/0/16 detail
Thu Oct 29 10:09:25.586 UTC

```

```

MKA Detailed Status for MKA Session
=====

```

```

Status: Secured - Secured MKA Session with MACsec

```

```

Local Tx-SCI           : fc58.9a05.9aa0/0001
Local Tx-SSCI         : 4
Interface MAC Address  : fc58.9a05.9aa0
MKA Port Identifier    : 1
Interface Name         : Hu0/3/0/16
CAK Name (CKN)        : 1234
CA Authentication Mode : PRIMARY-PSK
Keychain               : kc
Member Identifier (MI) : C45D5F2028232022F30C6BD8
Message Number (MN)   : 9427
Authenticator         : NO
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)   : C45D5F2028232022F30C6BD800000036 (54)
Old SAK Status        : No Rx, No Tx
Old SAK AN            : 0
Old SAK KI (KN)      : RETIRED (53)

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       : ms
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      : 3
# of MACsec Capable Live Peers Responded : 3

```

```

# of MACSec Suspended Peers        : 0

```

Live Peer List:

```

-----
          MI              MN              Rx-SCI              SSCI  KS-Priority
-----
499BFCA3044D65A9BA4FC219    9427    fc58.9a05.9aa8/0001    3      16
56765F41D6BE434860E62991    9427    fc58.9a05.9ab0/0001    2      16
7236084C87ADD66D59C63FE1    9425    fc58.9a05.9ab1/0001    1      16

```

Potential Peer List:

```

-----
          MI              MN              Rx-SCI              SSCI  KS-Priority
-----

```

Suspended Peer List:

```

-----
          Rx-SCI          SSCI
-----

```

Peers Status:

```

Last Tx MKPDU      : 2020 Oct 29 10:09:25.071
Peer Count        : 3

```

```

RxSCI              : FC589A059AB00001
MI                 : 56765F41D6BE434860E62991
Peer CAK           : Match
Latest Rx MKPDU    : 2020 Oct 29 10:09:24.072

```

```

RxSCI              : FC589A059AA80001
MI                 : 499BFCA3044D65A9BA4FC219
Peer CAK           : Match
Latest Rx MKPDU    : 2020 Oct 29 10:09:25.574

```

```

RxSCI              : FC589A059AB10001
MI                 : 7236084C87ADD66D59C63FE1
Peer CAK           : Match
Latest Rx MKPDU    : 2020 Oct 29 10:09:24.572

```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the interface and other MACsec parameters.

Step 5 Verify the MACsec session counter statistics.

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

```


The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any.

This completes the verification of MACsec encryption on the IOS-XR.

Verifying MACsec Encryption on Cisco 8000 Series Routers

MACsec encryption on the router hardware can be verified by running relevant commands in the Privileged Executive Mode.

To verify if MACsec encryption has been correctly configured, follow these steps.

SUMMARY STEPS

1. Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.
2. To verify if the hardware programming is done, use the following command:

DETAILED STEPS

Step 1 Verify the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

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    : FALSE
  Sectag offset      : 0
  Rx SC 1
  Rx SCI              : 008a962d74000001
  Peer MAC            : 00:8a:96:2d:74:00
  Stale                : NO
  SAK Data
  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]              : ***
  SAK Len              : 16
  SAK Version         : 1
  HashKey[0]          : ***
  HashKey Len         : 16
  Conf offset         : 30
  Cipher Suite        : GCM-AES-XPB-128
  CtxSalt[0]          : 01 8f 2f 0c 63 ff 2e d6 a5 bf 27 0e
  ssci                 : 1
  Tx SA Program Req[0]: 2019 Oct 08 07:37:14.908
  Tx SA Program Rsp[0]: 2019 Oct 08 07:37:14.931

```

When more than 1 RX SA is configured in P2MP networks, the output would be as follows:

```

Router#show macsec ea idb interface Hu0/3/0/16
Thu Oct 29 10:10:10.947 UTC

IDB Details:
  if_sname             : Hu0/3/0/16
  if_handle            : 0x1800240
  MacSecControlledIfh : 0x1800270
  MacSecUnControlledIfh : 0x1800278
  Replay window size  : 64
  Local MAC           : fc:58:9a:05:9a:a0
  Rx SC Option(s)     : Validate-Frames Replay-Protect
  Tx SC Option(s)     : Protect-Frames Always-Include-SCI
  Security Policy      : MUST SECURE
  Delay Protection     : FALSE
  Sectag offset       : 0
  Rx SC 1
    Rx SCI              : fc589a059ab10001
    Peer MAC            : fc:58:9a:05:9a:b1
    Stale                : NO
    SAK Data
      SAK[1]            : ***
      SAK Len            : 32
      SAK Version       : 3
      HashKey[1]        : ***
      HashKey Len       : 16
      Conf offset       : 0
      Cipher Suite      : GCM-AES-XPB-256
      CtxSalt[1]        : c4 6b 5f 21 28 23 20 22 f3 0c 6b d8
      ssci               : 1
      Rx SA Program Req[1]: 2020 Oct 29 05:04:30.803
      Rx SA Program Rsp[1]: 2020 Oct 29 05:04:30.807
  Rx SC 2
    Rx SCI              : fc589a059ab00001
    Peer MAC            : fc:58:9a:05:9a:b0
    Stale                : NO
    SAK Data
      SAK[1]            : ***
      SAK Len            : 32
      SAK Version       : 3
      HashKey[1]        : ***
      HashKey Len       : 16
      Conf offset       : 0
      Cipher Suite      : GCM-AES-XPB-256
      CtxSalt[1]        : c4 6b 5f 22 28 23 20 22 f3 0c 6b d8

```

```

      ssci                : 2
      Rx SA Program Req[1]: 2020 Oct 29 05:04:30.792
      Rx SA Program Rsp[1]: 2020 Oct 29 05:04:30.796

Rx SC 3
Rx SCI                : fc589a059aa80001
Peer MAC              : fc:58:9a:05:9a:a8
Stale                 : NO
SAK Data
  SAK[1]              : ***
  SAK Len              : 32
  SAK Version          : 3
  HashKey[1]          : ***
  HashKey Len         : 16
  Conf offset         : 0
  Cipher Suite        : GCM-AES-XPN-256
  CtxSalt[1]         : c4 6b 5f 23 28 23 20 22 f3 0c 6b d8
  ssci                : 3
  Rx SA Program Req[1]: 2020 Oct 29 05:04:30.788
  Rx SA Program Rsp[1]: 2020 Oct 29 05:04:30.792

Tx SC
Tx SCI                : fc589a059aa00001
Active AN             : 1
Old AN                : 0
Next PN               : 1, 1, 1, 1
SAK Data
  SAK[1]              : ***
  SAK Len              : 32
  SAK Version          : 3
  HashKey[1]          : ***
  HashKey Len         : 16
  Conf offset         : 0
  Cipher Suite        : GCM-AES-XPN-256
  CtxSalt[1]         : c4 6b 5f 24 28 23 20 22 f3 0c 6b d8
  ssci                : 4
  Tx SA Program Req[1]: 2020 Oct 29 05:04:32.773
  Tx SA Program Rsp[1]: 2020 Oct 29 05:04:32.780

```

The **if_handle** field provides the IDB instance location.

The **Replay window size** field displays the configured window size.

The **Security Policy** field displays the configured security policy.

The **Local MAC** field displays the MAC address of the router.

The **Peer MAC** field displays the MAC address of the peer. This confirms that a peer relationship has been formed between the two routers.

Step 2 To verify if the hardware programming is done, use the following 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-XPB-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 Ingress
Direction        : Ingress
Conf Offset      : 00000030
Drop Type        : 0x00000002
SA In Use        : YES
ReplayProtect    : YES
lowPN            : 1
nxtPN            : 2

```

This completes the verification of MACsec encryption on the router hardware, and the configuration and verification of MACsec encryption.

MACsec SecY Statistics

The following methods are used to query MACsec SecY statistics such as, encryption, decryption, and the hardware statistics.

- CLI
- SNMP MIB

Querying SNMP Statistics Using CLI

The following example shows how to query SNMP statistics using a CLI. Use the **show macsec secy statistics interface *interface name*** command to display the MACsec SecY statistics details.

```

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

```

MACsec SNMP MIB (IEEE8021-SECY-MIB)

The IEEE8021-SECY-MIB provides Simple Network Management Protocol (SNMP) access to the MAC security entity (SecY) MIB running with IOS XR MACsec-enabled line cards. The IEEE8021-SECY-MIB is used to query on the SecY data, encryption, decryption, and the hardware statistics. The SecY MIB data is queried only on the Controlled Port.

The object ID of 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 3: IEEE8021-SECY-MIB Table

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 information, see the SecY IEEE MIB at the following URL:

<http://www.ieee802.org/1/files/public/MIBs/IEEE8021-SECY-MIB-200601100000Z.mib>

secyIfTable

The following table represents the system level information for each interface supported by the MAC security entity. The index tuple for this table is secyIfInterfaceIndex.

Table 4: secyIfTable

Object	Object identifier
secyIfInterfaceIndex	1.0.8802.1.1.3.1.1.1.1.1
secyIfMaxPeerSCs	1.0.8802.1.1.3.1.1.1.1.2
secyIfRxMaxKeys	1.0.8802.1.1.3.1.1.1.1.3
secyIfTxMaxKeys	1.0.8802.1.1.3.1.1.1.1.4
secyIfProtectFramesEnable	1.0.8802.1.1.3.1.1.1.1.5
secyIfValidateFrames	1.0.8802.1.1.3.1.1.1.1.6
secyIfReplayProtectEnable	1.0.8802.1.1.3.1.1.1.1.7
secyIfReplayProtectWindow	1.0.8802.1.1.3.1.1.1.1.8
secyIfCurrentCipherSuite	1.0.8802.1.1.3.1.1.1.1.9
secyIfAdminPt2PtMAC	1.0.8802.1.1.3.1.1.1.1.10
secyIfOperPt2PtMAC	1.0.8802.1.1.3.1.1.1.1.11
secyIfIncludeSCIEnable	1.0.8802.1.1.3.1.1.1.1.12
secyIfUseESEnable	1.0.8802.1.1.3.1.1.1.1.13
secyIfUseSCBEnable	1.0.8802.1.1.3.1.1.1.1.14

secyTxSCTable

The following table provides information about the status of each transmitting SC supported by the MAC security entity. The index tuple for this table is secyIfInterfaceIndex.

Table 5: secyTxSCTable

Object	Object identifier
secyTxSCI	1.0.8802.1.1.3.1.1.2.1.1
secyTxSCState	1.0.8802.1.1.3.1.1.2.1.2
secyTxSCEncodingSA	1.0.8802.1.1.3.1.1.2.1.3
secyTxSCEncipheringSA	1.0.8802.1.1.3.1.1.2.1.4
secyTxSCCreatedTime	1.0.8802.1.1.3.1.1.2.1.5

Object	Object identifier
secyTxSCStartedTime	1.0.8802.1.1.3.1.1.2.1.6
secyTxSCStoppedTime	1.0.8802.1.1.3.1.1.2.1.7

secyTxSatable

The following table provides information about the status of each transmitting SA supported by the MAC security entity. The index tuple for this table is: {secyIfInterfaceIndex, secyTxSA}.

Table 6: secyTxSatable

Object	Object identifier
secyTxSA	1.0.8802.1.1.3.1.1.3.1.1
secyTxSAState	1.0.8802.1.1.3.1.1.3.1.2
secyTxSANextPN	1.0.8802.1.1.3.1.1.3.1.3
secyTxSAConfidentiality	1.0.8802.1.1.3.1.1.3.1.4
secyTxSASAKUnchanged	1.0.8802.1.1.3.1.1.3.1.5
secyTxSACreatedTime	1.0.8802.1.1.3.1.1.3.1.6
secyTxSAStartedTime	1.0.8802.1.1.3.1.1.3.1.7
secyTxSAStoppedTime	1.0.8802.1.1.3.1.1.3.1.8

secyRxSCTable

The following table provides information about the status of each receiving SC supported by the MAC security entity. The index tuple for this table is: {secyIfInterfaceIndex, secyRxSCI}.

Table 7: secyRxSCTable

Object	Object identifier
secyRxSCI	1.0.8802.1.1.3.1.1.4.1.1
secyRxSCState	1.0.8802.1.1.3.1.1.4.1.2
secyRxSCCurrentSA	1.0.8802.1.1.3.1.1.4.1.3
secyRxSCCreatedTime	1.0.8802.1.1.3.1.1.4.1.4
secyRxSCStartedTime	1.0.8802.1.1.3.1.1.4.1.5
secyRxSCStoppedTime	1.0.8802.1.1.3.1.1.4.1.6

secyRxSatable

The following table provides information about the status of each receiving SA supported by the MAC security entity. The index tuple for this table is: {secyIfInterfaceIndex, secyRxSCI, secyRxSA}.

Table 8: secyRxSatable

Object	Object identifier
secyRxSA	1.0.8802.1.1.3.1.1.5.1.1
secyRxSAState	1.0.8802.1.1.3.1.1.5.1.2
secyRxSASNextPN	1.0.8802.1.1.3.1.1.5.1.3
secyRxSASAKUnchanged	1.0.8802.1.1.3.1.1.5.1.4
secyRxSACreatedTime	1.0.8802.1.1.3.1.1.5.1.5
secyRxSASStartedTime	1.0.8802.1.1.3.1.1.5.1.6
secyRxSASStoppedTime	1.0.8802.1.1.3.1.1.5.1.7

secyCipherSuiteTable

The following table is a list of selectable cipher suites for the MAC security entity. The index tuple for this table is: {secyCipherSuiteIndex}.

Table 9: secyCipherSuiteTable

Object	Object identifier
secyCipherSuiteIndex	1.0.8802.1.1.3.1.1.6.1.1
secyCipherSuiteId	1.0.8802.1.1.3.1.1.6.1.2
secyCipherSuiteName	1.0.8802.1.1.3.1.1.6.1.3
secyCipherSuiteCapability	1.0.8802.1.1.3.1.1.6.1.4
secyCipherSuiteProtection	1.0.8802.1.1.3.1.1.6.1.5
secyCipherSuiteProtectionOffset	1.0.8802.1.1.3.1.1.6.1.6
secyCipherSuiteDataLengthChange	1.0.8802.1.1.3.1.1.6.1.7
secyCipherSuiteICVLength	1.0.8802.1.1.3.1.1.6.1.8
secyCipherSuiteRowStatus	1.0.8802.1.1.3.1.1.6.1.9

secyTxSASStatsTable

The following table that contains the statistics objects for each transmitting SA in the MAC security entity.

Table 10: secyTxSAStatsTable

Object	Object identifier
secyTxSAStatsProtectedPkts	1.0.8802.1.1.3.1.2.1.1.1
secyTxSAStatsEncryptedPkts	1.0.8802.1.1.3.1.2.1.1.2
secyTxSCStatsProtectedPkts	1.0.8802.1.1.3.1.2.2.1.1
secyTxSCStatsEncryptedPkts	1.0.8802.1.1.3.1.2.2.1.4
secyTxSCStatsOctetsProtected	1.0.8802.1.1.3.1.2.2.1.10
secyTxSCStatsOctetsEncrypted	1.0.8802.1.1.3.1.2.2.1.11

secyTxSCStatsTable

The following table that contains the statistics objects for each transmitting SC in the MAC security entity.

Table 11: secyTxSCStatsTable

Object	Object identifier
secyTxSCStatsProtectedPkts	1.0.8802.1.1.3.1.2.2.1.1
secyTxSCStatsEncryptedPkts	1.0.8802.1.1.3.1.2.2.1.4
secyTxSCStatsOctetsProtected	1.0.8802.1.1.3.1.2.2.1.10
secyTxSCStatsOctetsEncrypted	1.0.8802.1.1.3.1.2.2.1.11

secyRxSAStatsTable

The following table that contains the statistics objects for each receiving SA in the MAC security entity.

Table 12: secyRxSAStatsTable

Object	Object identifier
secyRxSAStatsUnusedSAPkts	1.0.8802.1.1.3.1.2.3.1.1
secyRxSAStatsNoUsingSAPkts	1.0.8802.1.1.3.1.2.3.1.4
secyRxSAStatsNotValidPkts	1.0.8802.1.1.3.1.2.3.1.13
secyRxSAStatsInvalidPkts	1.0.8802.1.1.3.1.2.3.1.16
secyRxSAStatsOKPkts	1.0.8802.1.1.3.1.2.3.1.25

secyRxSCStatsTable

The following table that contains the statistics objects for each receiving SC in the MAC security entity.

Table 13: *secyRxSCStatsTable*

Object	Object identifier
secyRxSCStatsUnusedSAPkts	1.0.8802.1.1.3.1.2.4.1.1
secyRxSCStatsNoUsingSAPkts	1.0.8802.1.1.3.1.2.4.1.2
secyRxSCStatsLatePkts	1.0.8802.1.1.3.1.2.4.1.3
secyRxSCStatsNotValidPkts	1.0.8802.1.1.3.1.2.4.1.4
secyRxSCStatsInvalidPkts	1.0.8802.1.1.3.1.2.4.1.5
secyRxSCStatsDelayedPkts	1.0.8802.1.1.3.1.2.4.1.6
secyRxSCStatsUncheckedPkts	1.0.8802.1.1.3.1.2.4.1.7
secyRxSCStatsOKPkts	1.0.8802.1.1.3.1.2.4.1.8
secyRxSCStatsOctetsValidated	1.0.8802.1.1.3.1.2.4.1.9
secyRxSCStatsOctetsDecrypted	1.0.8802.1.1.3.1.2.4.1.10

secyStatsTable

The following table lists the objects for the statistics information of each Secy supported by the MAC security entity.

Table 14: *secyStatsTable*

Object	Object identifier
secyStatsTxUntaggedPkts	1.0.8802.1.1.3.1.2.5.1.1
secyStatsTxTooLongPkts	1.0.8802.1.1.3.1.2.5.1.2
secyStatsRxUntaggedPkts	1.0.8802.1.1.3.1.2.5.1.3
secyStatsRxNoTagPkts	1.0.8802.1.1.3.1.2.5.1.4
secyStatsRxBadTagPkts	1.0.8802.1.1.3.1.2.5.1.5
secyStatsRxUnknownSCIPkts	1.0.8802.1.1.3.1.2.5.1.6
secyStatsRxNoSCIPkts	1.0.8802.1.1.3.1.2.5.1.7
secyStatsRxOverrunPkts	1.0.8802.1.1.3.1.2.5.1.8

Obtaining the MACsec Controlled Port Interface Index

The ifindex of the controlled port can be obtained using the following commands:

- **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

In the following examples, it is assumed that the configured SNMP community is public, and the management IP of the box is 10.0.0.1.

To perform SNMP walk on the entire SECY MIB for the router, use the following command:

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

To query on the secyTxSCTable to get the TxSCI for interface Gi0/1/0/0, using the ifindex of MACSecControlled0/1/0/0 that is 18, use the following command:

```
snmpget -v2c -c public 10.0.0.1 iso.0.8802.1.1.3.1.1.2.1.18
```

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

The Cisco IOS XR Software Release 7.0.14 introduces the support for power-on self-test (POST) known answer test (KAT) for common criteria and FIPS compliance for the MACsec-enabled hardware on Cisco 8000 Series Routers. In POST KAT, the KAT is executed immediately after the cipher module is powered on. With this feature enabled, the system allows the traffic to pass through the MACsec-enabled hardware only if it passes the KAT. If the KAT fails, the modules shut down and the ports do not come up.

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

How to Configure Power-on Self-Test KAT

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 in place, 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 powered up.

**Note**

- With KAT enabled, you can expect a delay of approximately 2 to 3 minutes for the boot up of a line card, in comparison to the boot up time without enabling KAT.
- If power-on self-test (POST) known answer test (KAT) is already enabled on the PHY, then the system does not allow you to configure the **hw-module macsec-fips-post location all** command again. This restriction is in place to prevent conflicts in configuration, especially in a configuration restore scenario. In such scenarios, you can make use of the **show hw-module macsec-mode fips-post** command to know of the respective running configurations in place.

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 LC boot sequence. The subsequent boot (based on the FIPS mode) state re-triggers the KAT. If there are multiple PHYs hardware in a module, then the system performs the KAT on each of the PHYs and returns the KAT results. If all PHYs pass the KAT, then 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 of the PHYs registers a KAT failure, then the module enters into an ERROR state and the system displays a critical ERROR SYSLOG output which reads as: *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 any 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.

Prerequisites for Power-on Self-Test KAT

- The k9sec package must be installed on the router.
- FIPS must be supported and enabled on the line card.

Configuration Example

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

Running Configuration

```
hw-module macsec-fips-post location 0/4/CPU0
!
```

Verification

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

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
```

Related Topics

- [Power-on Self-Test KAT for Common Criteria and FIPS, on page 44](#)

Associated Commands

- [hw-module macsec-fips-post](#)
- [show hw-module macsec-fips-post](#)

Related Commands for MACsec

The following commands are available to verify the SNMP results.

Command	Description
show macsec mka session detail	Displays the details of all MACsec Key Agreement (MKA) sessions on the device.
show macsec mka interface detail	Verifies the MACsec MKA status on the interface.
show macsec ea idb interface	Verifies the MACsec encryption and hardware interface descriptor block (IDB) information on the interface.

