

Configure MACSec

This module describes how to configure Media Access Control Security (MACSec) encryption on the ASR 9000 Series Aggregation Services Routers. MACSec is a Layer 2 IEEE 802.1AE standard for encrypting packets between two MACSec-capable routers.

Feature History for Configure MACSec

Release	Modification
Release 5.3.2	This feature was introduced.
Release 6.0.1	This feature was modified to support VLAN sub-interfaces and bundles.
Release 6.1.2	This feature was modified to introduce MACsec as a service.
Release 6.3.3	Introduced the support for global MACsec shutdown.
Release 6.3.3	Introduced the support for MACsec SAK rekey interval.
Release 6.5.1	MACSec support was introduced on Cisco ASR 9901 Routers.
Release 6.6.1	A9K-MPA-32x1GE MPA card was introduced with MACSec support for Cisco IOS XR.
Release 6.6.2	MACSec support with A9K-MPA-32x1GE extended to IOS XR 64-bit.
Release 7.1.1	MACsec ISSU feature was introduced for 64-bit IOS XR.
Release 7.1.3	MACSec support was introduced on Cisco ASR 9000 5th generation line cards, Cisco ASR 9903 1.6T chassis and Cisco ASR 9903 2T port expansion card running Cisco IOS XR 64-bit.

- Understanding MACsec Encryption, on page 2
- Advantages of Using MACsec Encryption, on page 3
- Types of MACsec Implementation, on page 3
- Configure the EAP-derived MACsec CAK length, on page 4
- MKA Authentication Process, on page 5
- Hardware Support for MACSec, on page 6
- MACSec Limitations for Cisco ASR 9901 Routers, on page 8
- MACsec PSK, on page 9
- PQC key exchange groups for EAP-TLS, on page 9

- Fallback PSK, on page 13
- WAN MACsec, on page 14
- Configuring and Verifying MACSec Encryption , on page 17
- Configuring and Verifying MACsec Encryption as a Service, on page 54
- Quantum safe key distribution options for MACsec, on page 78
- Understanding SKIP, on page 86
- Global MACsec Shutdown, on page 92
- MACsec ISSU, on page 94
- MACsec SNMP MIB (IEEE8021-SECY-MIB), on page 100

Understanding MACsec Encryption

Security breaches can occur at any layer of the OSI model. At Layer 2, some of the common breaches at Layer 2 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 Customer Edge (CE) router interfaces that connect to Provider Edge (PE) routers and on all the provider router interfaces.

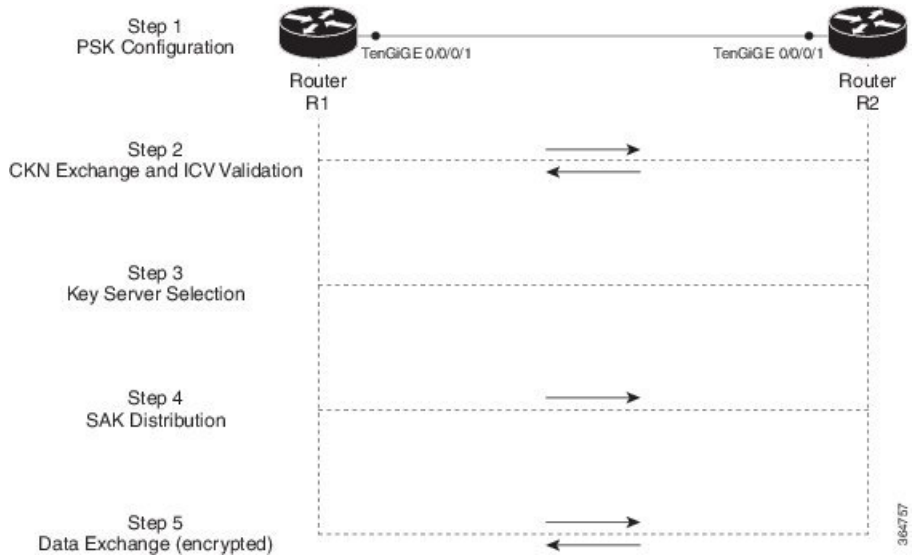
MACservice can be deployed in the network as a technology or as a service. For more information, see Types of MACsec Implementation, on page 3

MACsec Authentication Process

MACsec provides encryption using Advanced Encryption Standard (AES) algorithm at the Layer 2. 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: MACsec 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). SAKs are generated for every data exchange between the peers.

Step 5: Encrypted data is exchanged between the peers.

Advantages of Using MACsec Encryption

- **Client-Oriented Mode**: MACsec is used in setups where two routers that are peering with each other can alternate as a key server or a key client prior to exchanging keys. The key server generates and maintains the CAK between the two peers.
- **Data Integrity Check**: MACsec uses MKA to generate an Integrity Check Value (ICV) for the frame arriving on the port. If the generated ICV is the same as the ICV in the frame, then the frame is accepted; otherwise it is dropped.
- **Data Encryption**: MACsec provides port-level encryption on the line card of the router. This means that the frames sent out of the configured port are encrypted and frames received on the port are decrypted. MACsec also provides a mechanism where you can configure whether only encrypted frames or all frames (encrypted and plain) are accepted on the interface.
- **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.

Types of MACsec Implementation

MACsec is implemented in the following ways:

- **MACsec** where it serves as an encryption method for all traffic on Ethernet links.

For more information on configuring MACsec, see Configuring and Verifying MACSec Encryption , on page 17.

For insights into deployment scenarios, see WAN MACsec, on page 14 .

- **MACsec as a service** where it serves as an encryption method for L2VPN and L3VPN traffic over a provider network. It provides a mechanism to provide encryption or decryption service for selected traffic across the WAN core. For example: aservice provider can charge encryption of voice calls at a premium. This solution supports both Point-to-Point as well as Multipoint service for all the traffic on the network.

  For more information on configuring MACsec as a service, see Configuring MACsec as a Service, on page 56

Both MACsec and MACsec service are mutually exclusive and can be deployed in the same network.

Configure the EAP-derived MACsec CAK length

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

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

Before you begin

Identify the MACsec policy used by the MACsec EAP session.

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

Procedure

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

Example:

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

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

Example:

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

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

What to do next

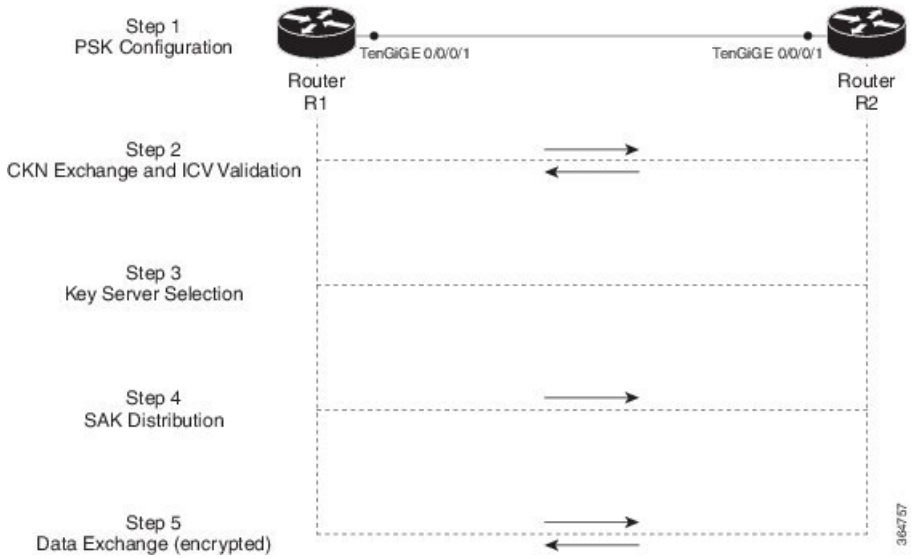
Configure the same EAP-derived CAK length in the MACsec policy on the peer.

MKA Authentication Process

MACsec provides encryption at the Layer 2, which is provided by the Advanced Encryption Standard (AES) algorithm that replaces the DES 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 2: 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.

- **Security tag**: 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.

Figure 3: MACsec Frame Format



Hardware Support for MACSec

The MACSec support on ASR 9000 Series Routers is compatible with the following chassis, line cards (LCs), and modular port adapters (MPAs).

Cisco IOS XR Software Release 7.3.2 and Release 7.4.1 introduce MACSec on sub-interfaces of ASR 9000 5th Generation Line Cards. For detailed list of supported PIDs, see the section, *Supported Line Cards for MACSec*.

Supported Chassis for MACSec

Table 1: Supported Chassis for MACSec

Chassis Type	Introduced Release for MACSec Support
Cisco ASR 9903 Router (with removable A9903-8HG-PEC card)	Release 7.4.1
Cisco ASR 9902 Router	Release 7.4.1
Cisco ASR 9903 Router (1.6T Fixed Board only or with removable A9903-20HG-PEC card)	Release 7.1.3
Cisco ASR 9901 Router	Release 6.5.1

Supported Modular Port Adapters for MACSec

The MACSec technology is supported on modular line cards when used with the following MPAs:

Table 2: Supported MPAs for MACSec

Hardware PIDs	Hardware Description	Introduced Release for MACSec Support
A9K-MPA-32X1GE	32-port GE Modular Port Adapter	Release 6.6.1
A9K-MPA-20X10GE	20-port 10 Gigabit Modular Port Adapter	Release 6.1.2
A9K-MPA-1X100GE	1-port 100 Gigabit Modular Port Adapter	Release 6.1.2
A9K-MPA-2X100GE	2-port 100 Gigabit Modular Port Adapter	Release 6.1.2

Supported Line Cards and Port Expansion Cards for MACSec

Following line cards and port expansion cards support MACSec:

Table 3: Supported Line Cards for MACSec

Line Card	Introduced Release for MACSec Support
200G and 400G modular line cards with A9K-MPA-20X10GE, A9K-MPA-1X100GE and A9K-MPA-2X100GE	Release 6.1.2
200G and 400G modular line cards with A9K-MPA-32X1GE	Release 6.6.1
4X100 GE and 8X100 GE OTN Line Card	Release 6.1.2
Cisco ASR 9000 Series 400-Gbps IPoDWDM Line Card - A9K-400G-DWDM-TR	Release 6.2.1
ASR 9000 5th Generation Line Cards	*See the table below for the list of supported PIDs and release information*

Table 4: Supported Port Expansion Cards for MACSec

Hardware PID	Hardware Description	Introduced Release for MACSec Support (on main interface)	Introduced Release for MACSec Support (on sub-interface)
A9903-8HG-PEC	ASR 9903 800G Multirate Port Expansion Card	Release 7.4.1	Release 7.4.1
A9903-20HG-PEC	ASR 9903 2T Multirate Port Expansion Card	Release 7.1.3	Release 7.3.2

Table 5: Supported ASR 9000 5th Generation Line Cards for MACSec

Hardware PID	Hardware Description	Introduced Release for MACSec Support (on main interface)	Introduced Release for MACSec Support (on sub-interface)
A9K-4HG-FLEX-X-SE	ASR 9000 400GE Service Edge Line Card Enhanced - 5th Generation	Release 25.2.1	Release 25.2.1
A99-4HG-FLEX-SE	ASR 9900 400GE Combo Service Edge Line Card - 5th Generation	Release 7.4.1	Release 7.4.1
A99-4HG-FLEX-TR	ASR 9900 400GE Combo Packet Transport Line Card - 5th Generation	Release 7.4.1	Release 7.4.1
A99-10X400GE-X-SE	ASR 9000 4T Service Edge Line Card - 5th Generation	Release 7.3.1	Release 7.3.2
A99-10X400GE-X-TR	ASR 9000 4T Packet Transport Line Card - 5th Generation	Release 7.3.1	Release 7.3.2
A9K-20HG-FLEX-SE	ASR 9000 2T Service Edge Combo Line Card - 5th Generation	Release 7.1.3	Release 7.3.2
A9K-20HG-FLEX-TR	ASR 9000 2T Packet Transport Combo Line Card - 5th Generation	Release 7.1.3	Release 7.3.2
A9K-8HG-FLEX-SE	ASR 9000 800G Service Edge Combo Line Card - 5th Generation	Release 7.1.3	Release 7.3.2
A9K-8HG-FLEX-TR	ASR 9000 800G Packet Transport Combo Line Card - 5th Generation	Release 7.1.3	Release 7.3.2

Note

- MACSec is not supported on ASR9000 24-port dual-rate 10G/1G service edge–optimized line card (A9K-24X10GE-1G-SE).

MACSec Limitations for Cisco ASR 9901 Routers

The following MACSec limitations are applicable for Cisco ASR 9901 routers:

- 1 Gigabit Ethernet interface supports MACSec only for GCM-AES-128 cipher.
- 1 Gigabit Ethernet interfaces created from 24 multi-rate ports do not support MACSec.
- MACSec on VLAN is not supported.
- Point-to-Multipoint scenarios are not supported.
- MACSec as a service is not supported.

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.

PQC key exchange groups for EAP-TLS

A PQC key exchange group is a TLS named group that

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

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
PQC-enabled MACsec with EAP-TLS	Release 26.3.1	Introduces PQC-capable TLS key exchange support to 802.1X EAP-TLS, with default CiscoSSL group advertisement and per-profile group configuration. This feature is supported on all MACsec-supported IOS XR platforms.

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

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

Note

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

Ensure compatibility and FIPS compliance for PQC EAP-TLS configuration

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

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

Supported PQC and classical key exchange groups

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

Table 7: Supported key exchange groups

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

Configure PQC key exchange groups for EAP-TLS

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

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

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

Before you begin

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

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

Procedure

Step 1 Enter EAP profile TLS configuration mode.

Example:

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

Step 2 Associate the EAP profile with the Dot1x profile.

Example:

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

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

Example:

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

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

Example:

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

Step 5 Display the configured groups in the EAP profile.

Example:

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

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

What to do next

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

Troubleshoot EAP-TLS key exchange group negotiation

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

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

Before you begin

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

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

Procedure

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

Step 2 Display detailed Dot1x information for the affected interface.

Example:

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

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

Step 3 Display the configured key exchange group list.

Example:

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

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

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

Example:

```
Router# debug eap method
```

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

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

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 25.

Active Fallback

The Cisco IOS XR Software Release 7.1.2 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 37 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.

WAN MACsec

MACsec services over the WAN or Metro Ethernet offers Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS).

WAN MACsec Use Cases

This section details the WAN MACsec use cases:

Use Case 1: MACSec in a L2VPN

The following figure illustrates the use of MACSec in a L2VPN network. In this topology, MACSec is configured on the PE-facing interfaces of the CE routers. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

In a L2VPN network that uses an Ethernet over MPLS (EoMPLS) pseudowire, the traffic between CE routers is encrypted by MACSec with VLAN tags in clear. The following figure illustrates the use of MACSec in a L2VPN cloud using an EoMPLS pseudowire. MACSec is configured on the PE-facing VLAN sub-interfaces of the CE router. The PE router encapsulates the MACSec frames with VLAN tags and MPLS labels in clear and sends the frames over the EoMPLS pseudowire.

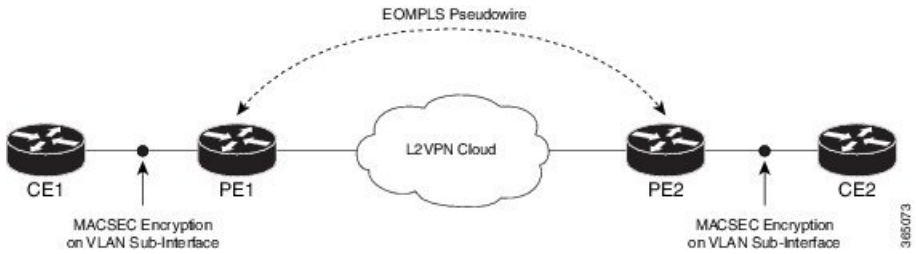
The following table lists the number of sub-interfaces with MACSec supported in a L2VPN.

Note To achieve scaling, sub-interfaces must be used.

Table 8: Supported MACSec Sessions on Sub-Interfaces

Interface Type	No. of Supported MACSec sessions (P2P)
10-GigE	5
40-GigE	21
100-GigE	42

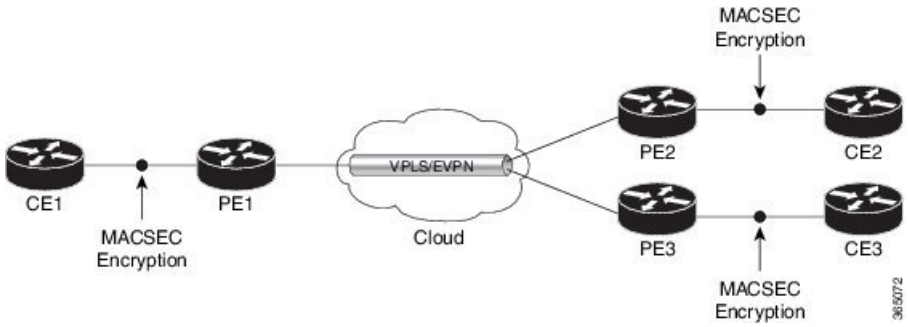
Figure 4: MACSec in a L2VPN Cloud



Use Case 2: 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. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces.

Figure 5: MACSec in a VPLS/EVPN Cloud



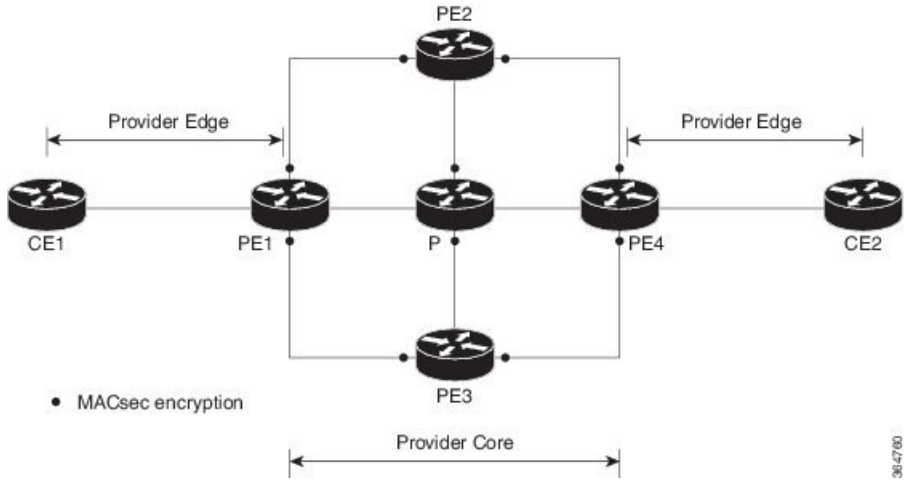
Use Case 3: MACSec in an MPLS Core Network

MACSec in an MPLS core network can be configured on physical interfaces, sub-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.

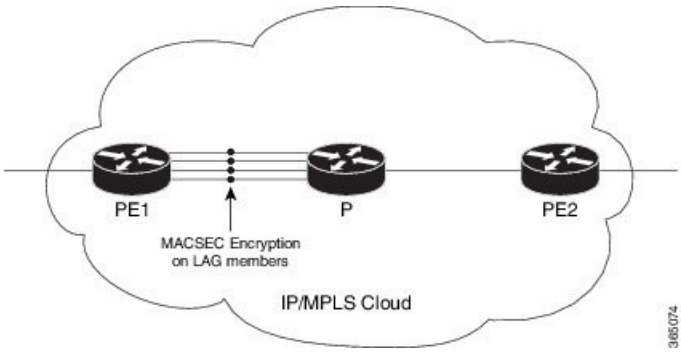
Figure 6: MACSec on Physical Interfaces in an MPLS Core Network



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/decryption takes place independently of other members in the group. MACSec can also be configured on VLAN sub-interfaces in these networks.

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

Figure 7: MACSec on a Link Bundle in an MPLS Core Network



MACsec Encryption on Layer 3 Subinterface

You can now implement MACsec on L3 subinterfaces to provide secure communication within a specific L3 VLAN. On implementing MACsec on the L3 subinterface, the MACsec encryption and authentication are unique to the traffic on that subinterface. As a result, you can control the traffic encryption for individual subinterfaces of a physical interface by customizing MACsec policies.

MACsec on L3 subinterface configurations are similar to the MACsec configurations on a physical interface. For a successful MACsec Key Agreement protocol (MKA) session to be up on any L3 subinterface, it must have a valid tagging protocol encapsulation and a VLAN identifier assigned. All L3 subinterfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined.

To configure MACsec Encryption on Layer 3 Subinterface, refer Configuring and Verifying MACsec Encryption on VLAN Subinterfaces , on page 28.

Guidelines and Restrictions for MACsec Encryption on Layer 3 Subinterface

- MACsec Encryption on the Layer 3 Subinterface is supported only in the N540-24Q8L2DD-SYS router.
- The L3 subinterfaces belonging to a physical interface must have either of the following encapsulation combinations:
  - 802.1Q with a single tag
  - 802.1Q with double tags
  - 802.1ad with a single tag
  - 802.1ad with double tags
- You must configure the same type of VLAN tag on all the subinterfaces belonging to a physical interface.
- The MACsec encryption on layer 3 subinterface supports VLAN identifier range of 1–4094.
- The encapsulation configured on the L3 subinterface and the number of VLAN tags in-clear configured on the associated MACsec policy must match. That is, if the encapsulation on the interface is 802.1Q or 802.1ad with a single tag, then the value of VLAN tags in-clear in the MACsec policy must be 1. Similarly, if the encapsulation on the interface is 802.1Q or 802.1ad with double tags, then the value of VLAN tags in-clear in the MACsec policy must be 2.

- MACsec support on physical interfaces and subinterfaces is mutually exclusive. To configure MACsec on subinterfaces, clear the MACsec configurations on the corresponding physical interface and conversely.
- The default VLAN tags in-clear value is 1.
- The following MACsec policy parameters must be identical in all subinterfaces in a physical interface:
  - security-policy
  - window-size
  - vlan-tags-in-clear
  - allow-lacp-in-clear
- MACsec on subinterfaces does not support data delay protection.
- We recommend keeping the MACsec session limit on anyline card or fixed port router, including all port-level and subinterface-level MACsec sessions, at 192 for optimal functioning of simultaneous hitless SAK rekey performance.

EAPoL Ether-Type and Destination-Address

In WAN MACsec, when two peers establish an MKA session using the standard EAPoL Ether-Type (0x888E) and destination MAC address (01:80:C2:00:00:03) via the service provider network, the Layer 2 intermediate devices may intercept and consume the EAPoL packets, which in turn can affect the MACsec session establishment between the two endpoints. To overcome this challenge, you can configure an alternate EAPoL Ether-Type, Destination MAC address, or both under the MACsec-enabled interface. For MACsec on subinterfaces, you can configure explicit Ether-Type and Destination MAC address under the subinterfaces; otherwise, the subinterfaces inherit the EAPoL configurations from the parent physical interface.

The alternate EAPoL Ether-Type suported is 0x876F. To configure an alternate EAPoL Ether-Type, refer Configure EAPoL Ether-Type 0x876F, on page 34.

The alternate EAPoL Destination MAC address supported is the multicast address FF:FF:FF:FF:FF or any nearest bridge group address. To configure an alternate EAPoL Destination-Address, refer Configure EAPoL Destination Address , on page 35.

Configuring and Verifying MACSec Encryption

MACSec can be configured on physical ethernet interfaces or VLAN sub-interfaces. The following section describes procedures for configuring and verifying MACSec configuration in any of the described deployment modes.

1. Creating a MACSec Key Chain.
2. Creating a MACSec Policy.
3. Applying MACSec on a Interface.

Creating a MACsec Key Chain

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 *infinite*.

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 7.1.2 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 7.1.2, 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 key IDs before upgrading to Release 7.1.2 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 7.1.2 or later, only one of these key IDs is retained. That is 'FF', the one that was applied second in this example.

SUMMARY STEPS

1. Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac_chain.
2. Enter the MACsec mode.
3. Provide a name for the MACsec key.
4. Enter the key string and the cryptographic algorithm to be used for the key.
5. Enter the validity period for the MACsec key (CKN) also known as the lifetime period.
6. Commit your configuration.

DETAILED STEPS

Procedure

Step 1 Enter the global configuration mode and provide a name for the MACsec keychain; for example, mac_chain.

Example:

```
RP/0/RSP0/CPU0:router(config)#key chain mac_chain
```

Step 2 Enter the MACsec mode.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_chain)#macsec
```

Step 3 Provide a name for the MACsec key.

The 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.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec)#key 1234abcd5678
```

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 inter-operate 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.

Step 4 Enter the key string and the cryptographic algorithm to be used for the key.

Example:

The key string is the CAK that is used for ICV validation by the MKA protocol.

```
! For AES 128-bit encryption

RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#key-string 12345678123456781234567812345678
 cryptographic-algorithm AES-128-CMAC

! For AES 256-bit encryption

RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#key-string
12345678123456781234567812345678123456781234567812345678123456781234567812345678 cryptographic
-algorithm AES-256-CMAC
```



Note In this example, we have used the AES 256-bit encryption algorithm, and therefore, the key string is 64 hexadecimal characters in length. 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.

Step 5 Enter the validity period for the MACsec key (CKN) also known as the lifetime period.

The lifetime period can be configured, with a duration in seconds, as a validity period between two dates (for example, Jan 01 2014 to Dec 31 2014), or with infinite validity.

The key is valid from the time you configure (in HH:MM:SS format). Duration is configured in seconds.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#lifetime 05:00:00 01
January 2015 duration 1800
```

An example of configuring the lifetime for a defined period:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#lifetime 05:00:00 20
```

```
february 2015 12:00:00 30 september 2015
```

An example of configuring the lifetime as infinite:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#lifetime
05:00:00 01 January 2015 infinite
```



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.

Step 6 Commit your configuration.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#commit
```

This completes the configuration of the MACsec keychain.

Creating a User-Defined MACsec Policy

SUMMARY STEPS

1. Enter the global configuration mode, and enter a name (mac_policy) for the MACsec policy.
2. Configure the cipher suite to be used for MACsec encryption.
3. Configure the confidentiality offset for MACsec encryption.
4. Enter the key server priority.
5. Configure the security policy parameters, either Must-Secure or Should-Secure.
6. Configure the replay protection window size.
7. Configure the ICV for the frame arriving on the port.
8. Commit your configuration and exit the global configuration mode.
9. Confirm the MACsec policy configuration.

DETAILED STEPS

Procedure

Step 1 Enter the global configuration mode, and enter a name (mac_policy) for the MACsec policy.

Example:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# macsec-policy mac_policy
```

Step 2 Configure the cipher suite to be used for MACsec encryption.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-XPN-256
RP/0/RSP0/CPU0:router(config-mac_policy)#GCM-AES-128
GCM-AES-256
GCM-AES-XPN-128
GCM-AES-XPN-256
```

In this example, the GCM-AES-XPN-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-XPN-256 encryption algorithm for higher data ports.



Note Use of non-XPN ciphers on high-speed links (40G and above) is not recommended, as frequent packet number exhaustion and rekey operations may lead to transient traffic loss.

Step 3 Configure the confidentiality offset for MACsec encryption.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
```



Note We recommend to change the offset value of the **conf-offset** <*offset_value*> command (MACsec encryption command) in the router 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.

Step 4 Enter the 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.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# key-server-priority 0
```

Step 5 Configure the security policy parameters, either Must-Secure or Should-Secure.

Must-Secure: Must-Secure imposes only MACsec encrypted traffic to flow. Hence, until MKA session is not secured, traffic will be dropped.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy must-secure
```

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

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# security-policy should-secure
```

Table 9: MACsec Security Policies

MKA		Secured MKA Session	Unsecured MKA Session
Security Policy	Must-secure	Encrypted traffic	Traffic drop (no Tx and no Rx)
	Should-secure	Encrypted traffic	Plain text or unencrypted traffic

Step 6 Configure the replay protection window size.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# window-size 64
```

This dictates the maximum out-of-sequence frames that are accepted. You can configure a value between 0 and 1024.

Step 7 Configure the ICV for the frame arriving on the port.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# include-icv-indicator
```

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.

Step 8 Commit your configuration and exit the global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router(config-mac_policy)# exit
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:router(config)# exit
```

Step 9 Confirm the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router# show running-config macsec-policy

macsec-policy mac_policy
conf-offset CONF-OFFSET-30
security-policy must-secure
window-size 64
cipher-suite GCM-AES-XPN-256
key-server-priority 0
include-icv-indicator
```

This completes the configuration of the MACsec policy.

Note

- Small packets might be dropped when Data Delay Protection (DDP) is enabled on many MACsec enabled interfaces of a scaled setup. To avoid this, enable DDP only on the interfaces which are absolutely necessary.
- For Cisco ASR 9000 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.

MACsec SAK Rekey Interval

From Cisco IOS XR Software Release 6.3.3 and later, 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

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.

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 P1
Router(config-macsec-policy-P1)#policy-exception lacp-in-clear
Router(config-macsec-policy-P1)#commit
```

Using the **allow** command:

```
Router#configure
Router(config)#macsec-policy P1
Router(config-macsec-policy-P1)#allow lacp-in-clear
Router(config-macsec-policy-P1)#commit
```

Running Configuration

With the **policy-exception** command:

```
Router#show run macsec-policy P1
macsec-policy P1
  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 P1
macsec-policy P1
  allow lacp-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**

Applying MACsec Configuration on an Interface

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.
- To enable MACsec on bundles:
  - Enable MACsec on all bundle members.
  - MACsec peers running IOS-XR version 24.1.1 or higher:
    - For routing protocols running on the bundle interface, configure **impose-overhead-on-bundle** in the MACsec policy to adjust the bundle interface MTU with MACsec overhead.
  - MACsec peers running IOS-XR versions prior to 24.1.1:
    - 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 ASR 9000 Series Routers*.

MACsec PSK Configuration on an Interface

```
Router#configure terminal
Router(config)#interface Te0/3/0/1/4
Router(config-if)#macsec psk-keychain kc policy mac_policy
```

To apply MACsec configuration on a physical interface without the MACsec policy, use the following command:

```
Router(config-if)#macsec psk-keychain 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 mac_policy
Router(config-if)#commit
```

Configuring and Verifying MACsec Encryption on Physical Interfaces

Enabling MACsec encryption on physical interfaces involves the following steps:

Configuration

1. Creating a MACsec Key Chain.
2. Creating a MACsec Policy.
3. Applying MACsec on a interface:

```
Router# configure
Router(config)# interface HundredGigE 0/5/0/16
Router(config-subif)# ipv4 address 192.168.16.1 255.255.255.0
Router(config-subif)# macsec psk-keychain kc fallback-psk-keychain fb
Router(config-subif)# commit
```

Running Configuration

Sub-Interface Configurations:

```
Router# show running-config interface HundredGigE 0/5/0/16
interface HundredGigE0/5/0/16
 ipv4 address 192.168.16.1 255.255.255.0
 macsec psk-keychain kc fallback-psk-keychain fb
 !
```

Verification

```
Router# show macsec mka summary
NODE: node0_5_CPU0
================================================================================================
   Interface-Name     Status     Cipher-Suite       KeyChain       PSK/EAP      CKN
================================================================================================
   Hu0/5/0/16     Secured  GCM-AES-XPN-256         kc          PRIMARY     1234
   Hu0/5/0/30     Secured  GCM-AES-XPN-256         kc          PRIMARY     1234
```

```
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
```

For detailed information on verifying MACsec encryption, refer Verifying MACsec Encryption on IOS XR, on page 37.

Configuring and Verifying MACsec Encryption on VLAN Subinterfaces

Enabling MACsec encryption on subinterfaces involves the following steps:

1. Creating a MACsec Key Chain.

2. Creating a MACsec Policy.
3. Applying MACsec on a Subinterface.

MACsec on VLAN Subinterfaces with Single Tag

Configuration

1. Creating a MACsec Key Chain:

```
Router# configure
Router(config)# key chain kc
Router(config-kc)# macsec
Router(config-kc-macsec)# key 1234
Router(config-kc-macsec-1234)# key-string
12345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
 aes-256-cmac
Router(config-kc-macsec-1234)# lifetime 05:00:00 1 January 2023 infinite
Router(config-kc-macsec-1234)# commit
```

2. Creating a MACsec Policy:

```
Router# configure
Router(config)# macsec-policy mp-SF1
RRouter(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
```

3. Applying MACsec on a Subinterface:

```
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
```

Running Configuration

MACsec Key Chain:

```
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:

```
Router# show running-config macsec-policy mp-SF1
macsec-policy mp-SF1
 ...
```

```
 vlan-tags-in-clear 1
!
```

Sub-Interface Configurations:

```
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
!
```

Verification

```
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
```

```
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       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      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
```

For detailed information on verifying MACsec encryption, refer Verifying MACsec Encryption on IOS XR, on page 37.

MACsec on VLAN Subinterfaces with Double Tag

Configuration

1. Creating a MACsec Key Chain:

```
Router# configure
Router(config)# key chain kc
Router(config-kc)# macsec
Router(config-kc-macsec)# key 1234
Router(config-kc-macsec-1234)# key-string
12345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
 aes-256-cmac
Router(config-kc-macsec-1234)# lifetime 05:00:00 1 January 2023 infinite
Router(config-kc-macsec-1234)# commit
```

2. Creating a MACsec Policy:

```
Router# configure
Router(config)# macsec-policy mp-SF1
RRouter(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
```

3. Applying MACsec on a Subinterface:

```
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
```

Running Configuration

MACsec Key Chain:

```
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:

```
Router# show running-config macsec-policy mp-SF2
macsec-policy mp-SF2
 ...
 vlan-tags-in-clear 2!
```

Subinterface Configurations:

```
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
```

Verification

```
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


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       : Must Secure
      Vlan Tags In Clear    : 2
      LACP In Clear         : FALSE
      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 : HundredGigE0/5/0/30.200
    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
```

For detailed information on verifying MACsec encryption, refer Verifying MACsec Encryption on IOS XR, on page 37.

Configure EAPoL Ether-Type 0x876F

Enabling EAPoL Ether-Type 0x876F involves the following steps:

Configuration

1. Creating a MACsec Key Chain.
2. (Optional) Creating a MACsec Policy.

3. Configure EAPoL ether-type.

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

4. Applying MACsec on a interface.

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

Running Configuration

```
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
!
```

Verification

```
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
```

Configure EAPoL Destination Address

Configuring EAPoL destination address involves the following steps:

Broadcast Address

The EAPoL destination address is set to broadcast address, FF:FF:FF:FF:FF to ensure the underlying L2 network will flood the EAPoL packets to all receivers.

Configuration

1. Creating a MACsec Key Chain.
2. (Optional) Creating a MACsec Policy.
3. Configure EAPoL destination address.

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

4. Applying MACsec on a interface.

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

Running Configuration

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

Verification

```
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
```

EAPoL Bridge Group Address

The EAPoL destination address can be set to the nearest bridge group address, for example 01:80:C2:00:00:00.

The following example shows EAPoL destination address configuration on a physical interface, which is inherited by the MACsec enabled subinterface.

Configuration

1. Creating a MACsec Key Chain.
2. (Optional) Creating a MACsec Policy.
3. Configure EAPoL destination address to a MACsec enabled physical interface.

```
Router(config)# interface HundredGigE0/1/0/1
Router(config-if)# eapol destination-address bridge-group-address 0180.c200.0000
Router(config-if)# commit
```

4. Configure MACsec on a subinterface.

```
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
```

Running Configuration

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

```
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
!
```

Verification

```
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
```

Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

Note With the introduction of active fallback functionality in Cisco IOS XR Software Release 7.1.2 (Release 6.7.2 for 32-bit Cisco IOS XR platforms), 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

Procedure

Step 1 Verify the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router#show macsec policy mac_policy

==========================================================================

Policy      Cipher         Key-Svr     Window  Conf

name        Suite          Priority    Size    Offset
```

```
================================================================================

mac_policy GCM-AES-XPN-256  0          64      30
```

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), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

Example:

Before the introduction of active fallback functionality:

```
RP/0/RSP0/CPU0:router#show macsec mka summary

NODE: node0_0_CPU0

=====================================================================
Interface     Status    Cipher Suite      KeyChain
=====================================================================

Fo0/0/0/1/0   Secured   GCM-AES-XPN-256   mac_chain

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

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0
=============================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server
=============================================================================
   Fo0/0/0/1/0        d46d.5023.3709/0001     1     Secured      YES


! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
=====================================================================
      Interface         Local-TxSCI       # Peers   Status  Key-Server
=====================================================================
  Fo0/0/0/1/1.8      e0ac.f172.4124/001d      1     Secured     Yes
```

With the introduction of active fallback functionality:

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

```
RP/0/RSP0/CPU0:router#show macsec mka summary

NODE: node0_0_CPU0
=============================================================================================
   Interface-Name     Status     Cipher-Suite       KeyChain       PSK/EAP      CKN
=============================================================================================
     Fo0/0/0/1/0      Secured  GCM-AES-XPN-256      mac_chain      PRIMARY      5555
     Fo0/0/0/1/0      Active   GCM-AES-XPN-256      mac_chain_fb   FALLBACK     5556

Total MACSec Sessions : 2
```

```
     Secured Sessions : 1
     Pending Sessions : 0
      Active Sessions : 1

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0
================================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
================================================================================================
   Fo0/0/0/1/0        d46d.5023.3709/0001     1      Secured      YES      PRIMARY     5555
   Fo0/0/0/1/0        d46d.5023.3709/0001     1      Active       YES     FALLBACK     5556
```

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 **Init**.

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

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

Example:

```
RP/0/RSP0/CPU0:router#show macsec mka session

NODE: node0_0_CPU0

================================================================
Interface    Local-TxSCI           # Peers   Status  Key-Server
================================================================

Fo0/0/0/1/0  001d.e5e9.aa39/0005      1      Secured  YES
```

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_1_CPU0
================================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
================================================================================================
     Fo0/0/0/1/0      001d.e5e9.aa39/0005     1      Secured      NO       PRIMARY     1234
     Fo0/0/0/1/0      001d.e5e9.aa39/0005     1      Active       NO       FALLBACK    1111
```

The **#Peers** field in the output confirms the presence of the peer you have configured on the physical interface, **Fo0/0/0/1/0**. 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.

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.



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
Fri May 28 07:18:45.726 UTC

NODE: node0_0_CPU0
==========================================================================================
Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
==========================================================================================
  Te0/0/0/1       6c8b.d34f.0635/0001     2      Secured      NO      FALLBACK     5556
```

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

Example:

Before the introduction of active fallback functionality:

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : 001d.e5e9.aa39/0005
Local Tx-SSCI           : 1
Interface MAC Address   : 001d.e5e9.aa39
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0
CAK Name (CKN)          : 1020000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A880BB45B9CE01584535F239
Message Number (MN)     : 5382
Authenticator           : NO
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : A880BB45B9CE01584535F23900000001 (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)
MKA Policy Name         : scale-21
Key Server Priority     : 20
Replay Window Size      : 40
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000001 (GCM-AES-128)
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
  ----------------------------------------------------------------------------
  4E33A276E7F79C04D80FE346       27114  d46d.5023.3704/0001     2         235
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
```

```
! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/1.8 detail
MKA Detailed Status for MKA Session
==================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4124/001d
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4124
MKA Port Identifier     : 29
Interface Name          : Fo0/0/0/1/1.8
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : 1EC4A4D1B0D75D3D5C2F6393
Message Number (MN)     : 1915
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 3
Latest SAK KI (KN)      : EB1E04894327E4EFA283C66200000003 (3)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (4)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : test12
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
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 (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  EB1E04894327E4EFA283C662        1908  001d.e5e9.b1c0/0037     2           0
Potential Peer List:
  MI                      MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
RP/0/RSP0/CPU0:macsec-CE1#sh macsec mka  interface Fo0/0/0/1/1.8
============================================================
Interface-name          KeyChain-Name           Policy Name
============================================================
Fo0/0/0/1/1.8           kc3                     test12


! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface FortyGigE0/0/0/1/0.1 detail
MKA Detailed Status for MKA Session
==================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4123/0001
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4123
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0.1
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A1DB3E42B4A543FBDBC281A6
```

```
Message Number (MN)     : 1589
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 1
Latest SAK KI (KN)      : AEC899297F5B0BDEF7C9FC6700000002 (2)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (1)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : mk_xpn1
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
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           : 2
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
  AEC899297F5B0BDEF7C9FC67         225  001d.e5e9.b1bf/0001     3           0
  0A4C49EE5B7401F1BECB7E22         147  001d.e5e9.f329/0001     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
```

With the introduction of active fallback functionality:

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

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/0/11 detail

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

Local Tx-SCI                    : 7061.7bea.1df4/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1df4
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/0/11
CAK Name (CKN)                  : 2111
CA Authentication Mode          : PRIMARY-PSK
Keychain                        : test1
Member Identifier (MI)          : 42A78BD6243539E917B8C6B2
Message Number (MN)             : 555
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)              : 69B39E87B3CBA673401E989100000001 (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                 : P12
Key Server Priority             : 20
Delay Protection                : TRUE
Replay Window Size              : 100
Include ICV Indicator           : TRUE
Confidentiality Offset          : 0
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
--------------------------------------------------------------------------------
69B39E87B3CBA673401E9891      617       008a.96d6.194c/0001    2        20

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

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:27:56.548
Peer Count             : 1

RxSCI                  : 008A96D6194C0001
  MI                    : 69B39E87B3CBA673401E9891
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 18 13:27:56.518

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

Local Tx-SCI                    : 7061.7bea.1df4/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1df4
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/0/11
CAK Name (CKN)                  : 2000
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : test1f
Member Identifier (MI)          : 1BB9428C721F6EE3E538C942
Message Number (MN)             : 553
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)              : 69B39E87B3CBA673401E989100000001 (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                 : P12
Key Server Priority             : 20
Delay Protection                : TRUE
Replay Window Size              : 100
Include ICV Indicator           : TRUE
Confidentiality Offset          : 0
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
------------------------------------------------------------------------------
8F59AD6021FA3E2D5F9E6231      615       008a.96d6.194c/0001    2        20

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

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:27:56.547
Peer Count             : 1

RxSCI                  : 008A96D6194C0001
  MI                   : 8F59AD6021FA3E2D5F9E6231
  Peer CAK             : Match
  Latest Rx MKPDU      : 2021 May 18 13:27:56.518

RP/0/RSP0/CPU0:router#
```

If sub-interfaces are configured, the output would be as follows. In this example, the status of FALLBACK-PSK is *Secured*.

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Hu0/0/0/0.6 detail
MKA Detailed Status for MKA Session
===================================
Status: Secured - Secured MKA Session with MACsec

Local Tx-SCI                    : 7061.7bea.1dc8/0006
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1dc8
MKA Port Identifier             : 6
Interface Name                  : Hu0/0/0/0.6
CAK Name (CKN)                  : 9999
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : D_tagf
Member Identifier (MI)          : 1DE18714A098B80964CC651E
Message Number (MN)             : 6203
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                   : 0
Latest SAK KI (KN)              : 1DE18714A098B80964CC651E00000001 (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               : 23510s
Time to exit suspension         : NA

MKA Policy Name                 : D_tag1
Key Server Priority             : 1
Delay Protection                : FALSE
Replay Window Size              : 1000
Include ICV Indicator           : TRUE
Confidentiality Offset          : 50
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

# of MACSec Suspended Peers              : 0

Live Peer List:
--------------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
--------------------------------------------------------------------------------
5C852D8F920306893D2BFB8F     10978      00c1.645f.2dd4/0006    2        11

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

Suspended Peer List:
--------------------------------------------------------------------------------
       Rx-SCI          SSCI
--------------------------------------------------------------------------------

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:29:15.687
Peer Count             : 1

RxSCI                  : 00C1645F2DD40006
  MI                    : 5C852D8F920306893D2BFB8F
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 18 13:29:15.769

RP/0/RSP0/CPU0:router#


! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/1/7 detail
Fri May 28 07:19:11.362 UTC

MKA Detailed Status for MKA Session
```

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

Local Tx-SCI                    : 6c8b.d34f.0635/0001
Local Tx-SSCI                   : 2
Interface MAC Address           : 6c8b.d34f.0635
MKA Port Identifier             : 1
Interface Name                  : Te0/0/0/1
CAK Name (CKN)                  : 5556
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : test2f
Member Identifier (MI)          : 6D14ECCDFB70E7E0463BD509
Message Number (MN)             : 20455
Authenticator                   : NO
Key Server                      : NO
MKA Cipher Suite                : AES-256-CMAC
Configured MACSec Cipher Suite  : GCM-AES-XPN-256

Latest SAK Status               : Rx & Tx
Latest SAK AN                   : 2
Latest SAK KI (KN)              : 1BBDDC0520C797C26AB7F1BF00000002 (2)
Old SAK Status                  : No Rx, No Tx
Old SAK AN                      : 1
Old SAK KI (KN)                 : RETIRED (1)

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           : 2
# of MACsec Capable Live Peers Responded : 0

Live Peer List:
------------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
------------------------------------------------------------------------------
1BBDDC0520C797C26AB7F1BF     19997      008a.96d6.194c/0001    3        16
B25B1000CC6FAE92D1F85738      139       dc77.4c3e.59c3/0001    1        16

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

Peers Status:
 Last Tx MKPDU          : 2021 May 28 07:19:10.153
 Peer Count             : 2

 RxSCI                  : 008A96D6194C0001
  MI                    : 1BBDDC0520C797C26AB7F1BF
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 07:19:09.960
```

```
 RxSCI                  : DC774C3E59C30001
  MI                    : B25B1000CC6FAE92D1F85738
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 07:19:10.180

RP/0/RSP0/CPU0:router#


RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/1/7.1 detail

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

Local Tx-SCI                    : 7061.7bff.e5e8/0001
Local Tx-SSCI                   : 2
Interface MAC Address           : 7061.7bff.e5e8
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/1/7.1
CAK Name (CKN)                  : 5556
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : test22f
Member Identifier (MI)          : 8FF3D1BBF09EA4AD6A0FC1B5
Message Number (MN)             : 81
Authenticator                   : NO
Key Server                      : YES
MKA Cipher Suite                : AES-256-CMAC
Configured MACSec Cipher Suite  : GCM-AES-XPN-256

Latest SAK Status               : Rx & Tx
Latest SAK AN                   : 3
Latest SAK KI (KN)              : 8FF3D1BBF09EA4AD6A0FC1B500000002 (2)
Old SAK Status                  : No Rx, No Tx
Old SAK AN                      : 2
Old SAK KI (KN)                 : RETIRED (1)

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               : 17930s
Time to exit suspension         : NA

MKA Policy Name                 : P123
Key Server Priority             : 10
Delay Protection                : FALSE
Replay Window Size              : 64
Include ICV Indicator           : FALSE
Confidentiality Offset          : 30
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           : 2
# of MACsec Capable Live Peers Responded : 2

# of MACSec Suspended Peers              : 0

Live Peer List:
--------------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
--------------------------------------------------------------------------------
6BCF91135F807CB9F57DDAAA       61       dc77.4c3e.5b05/0001    1        24
D81CFE93D07E932DDC33666E       44       00a7.4250.56c2/0001    3        25
```

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

Suspended Peer List:
-----------------------------------------------------------------------------
       Rx-SCI          SSCI
-----------------------------------------------------------------------------

Peers Status:
 Last Tx MKPDU          : 2021 May 28 13:16:50.992
 Peer Count             : 2

 RxSCI                  : DC774C3E5B050001
  MI                    : 6BCF91135F807CB9F57DDAAA
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 13:16:51.312

 RxSCI                  : 00A7425056C20001
  MI                    : D81CFE93D07E932DDC33666E
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 13:16:50.945
RP/0/RSP0/CPU0:router#
```

Step 5 Verify the MACsec session counter statistics.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0


MKA Statistics for Session on interface (Fo0/0/0/1/0)
=====================================================
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.......... 3
SAKs Rekeyed............ 2
SAKs Received........... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted...... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success.......... 5425
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
```

```
MKPDUs Rx Error..... 0
MKPDUs Rx Success..... 4932

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

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

! If sub-interfaces are configured, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8

MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
=======================================================
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..... 9
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted..... 1973
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 1965
      "Distributed SAK".. 9
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success..... 1973
   MKPDUs Tx Fail..... 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error..... 0
   MKPDUs Rx Success..... 1965
```

! In a VPLS network with a mulitpoint interface, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1

MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
=======================================================
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........... 2
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1608
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 406
      "Distributed SAK".. 2
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1608
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1802
```

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 ASR 9000

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. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

DETAILED STEPS

Procedure

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

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0


IDB Details:
if_sname : Fo0/0/0/1/0
```

```
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE
Sectag offset : 8
VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
 cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064
```

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

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname                : Fo0/0/0/1/0.1
  if_handle               : 0x2e40
  Replay window size      : 1024
  Local MAC               : e0:ac:f1:72:41:23
  Rx SC Option(s)         : Validate-Frames Replay-Protect
  Tx SC Option(s)         : Protect-Frames Always-Include-SCI
  Security Policy         : MUST SECURE
  Sectag offset           : 8
  VLAN                    : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                          : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)
  Rx SC 1
    Rx SCI                : 001de5e9f3290001
    Peer MAC              : 00:1d:e5:e9:f3:29
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
```

```
    Conf offset         : 50
    Cipher Suite        : GCM-AES-XPN-256
    CtxSalt[1]          : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67
  Rx SC 2
    Rx SCI              : 001de5e9b1bf0001
    Peer MAC            : 00:1d:e5:e9:b1:bf
    Stale               : NO
    SAK Data
      SAK[1]            : ***

      SAK Len           : 32
      HashKey[1]        : ***
      HashKey Len       : 16
      Conf offset       : 50
      Cipher Suite      : GCM-AES-XPN-256
      CtxSalt[1]        : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
  Tx SC
    Tx SCI              : e0acf17241230001
    Active AN           : 1
    Old AN              : 0
    Next PN             : 1, 1, 0, 0
    SAK Data
      SAK[1]            : ***

      SAK Len           : 32
      HashKey[1]        : ***
      HashKey Len       : 16
      Conf offset       : 50
      Cipher Suite      : GCM-AES-XPN-256
      CtxSalt[1]        : ae ca 99 28 7f 5b 0b de f7 c9 fc 67
```

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 Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
```

```
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff
```

Step 3 Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

Step 4 Verify the MACsec Secure Channel (SC) information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0


MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbffffb8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
      Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
```

```
Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

Configuring and Verifying MACsec Encryption as a Service

This section describes how MACsec can be implemented as a service in a L2VPN or L3VPN setup.

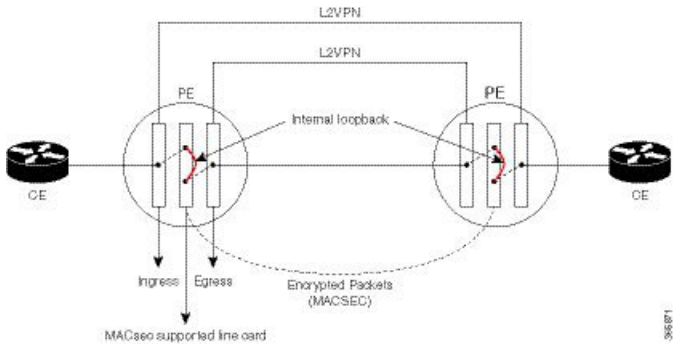
Note MACsec encryption is not supported on interface bundles, but is supported on member links .

Use Case 1: MACsec in an L2VPN Topology

In this topology, MACsec is configured on the PE router (with the interfaces facing the CE router) to provide crypto or encryption service on the PE router as a premium service for selected traffic on the WAN core. The interfaces can be physical ethernet interfaces or VLAN sub-interfaces. The customer can select the traffic that will be part of the encryption.

The following figure illustrates the use of MACsec as a service in an L2VPN network:

Figure 8: MACsec in an L2VPN topology



The data transferred between the CE router and the PE router are not encrypted. The data in clear format is sent to the access port of the PE router.

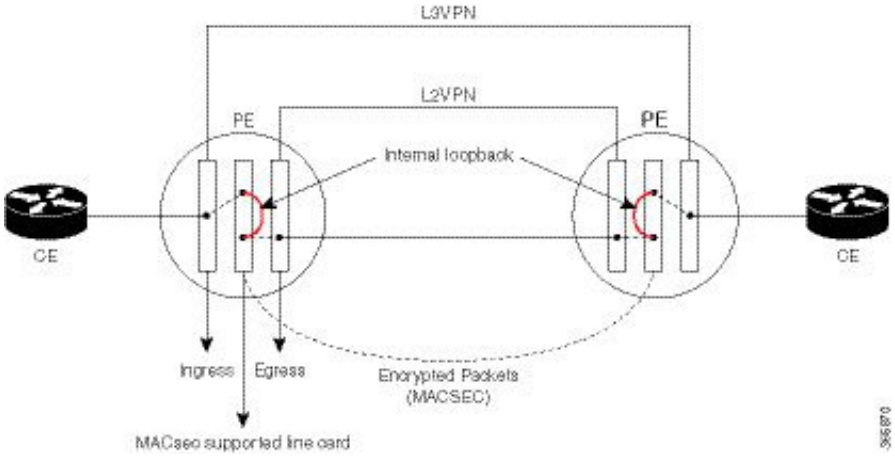
The PE router ports that receive traffic from CE routers divert the traffic using L2 local switching to the line card configured to perform encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE. After this, the packets are sent completely encrypted to the opposite PE router.

Use Case 2: MACsec in an L3VPN Topology

The following figure illustrates the use of MACsec as a service in an L3VPN environment. The topology is similar to an L2VPN set up where MACsec is configured on the PE router (where the interfaces facing the

CE router) to provide crypto or encryption services on the PE router as a premium service for selected traffic on the WAN core.

Figure 9:



The data transferred between the CE router and the PE router is not encrypted. The data is sent in clear-text format to the PE router access port. The PE router for each sub-interface distinguishes whether the data is part of MACsec encrypted service.

The PE router ports that receive traffic from CE routers divert the traffic using L3 local switching to the line card port configured to do encryption. The MACsec configuration creates internal loopback to the port configured for L2VPN to the opposite PE router. After this, the packets are sent completely encrypted to the opposite PE.

Restrictions

Ports usage for encryption on the line card must meet the following criteria:

- The ports must be TenGigE interfaces.
- Both the ports must belong either to an A9K-MPA-20X10GE MPA, or they must be breakout interfaces from one of the A9K-8X100GE-SE, A9K-8X100GE-TR, A9K-4X100GE-SE, or A9K-4X100GE-TR line cards.
- If the interfaces belong to A9K-MPA-20x10GE line card, then both the interfaces must be either in port range 0-9, or in port range 10-19. One interface from range 0-9 and other from 10-19 must not be selected.
- If the interfaces are breakout interfaces, then both of them must belong to the same HundredGigE port.

Note These restrictions apply only to MACsec interfaces. These restrictions do not apply to the CE or core-facing interfaces.

Configuring MACsec as a Service

SUMMARY STEPS

1. Enter interface configuration mode.
2. Configure the MACsec service.
3. Commit your configuration and exit global configuration mode.
4. Confirm the MACsec policy configuration.

DETAILED STEPS

Procedure

Step 1 Enter interface configuration mode.

Example:

```
RP/0/RSP/CPU0:router# interface <interface> 15.10 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 10
```

Step 2 Configure the MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-subif)# macsec-service decrypt-port <intf>17.10 psk-keychain
<keychain_name> [policy <macsec_policy>]
```

Step 3 Commit your configuration and exit global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router# commit
RP/0/RSP0/CPU0:router# exit
```

Step 4 Confirm the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router#

show running-config interface <interface> 15.10

interface <interface> 15.10
 macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
 encapsulation dot1q 10
```

Configuring MACsec Service for L2VPN Network

Configuring the MACsec service for L2VPN network, involves the following steps:

SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router.
3. Enable MACsec service.
4. Configure service port.
5. Configure the Xconnect group between ports.
6. Connect the ports.

DETAILED STEPS

Procedure

Step 1 Enter global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Step 2 Enter interface configuration mode and configure port facing the CE router.

The interface can be a physical interface or a VLAN sub-interface.

Example:

```
RP/0/RSP0/CPU0:router(config)# interface  <interface>15.10  l2transport
 encapsulation dot1q 10
```

Step 3 Enable MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface  <interface>16.10  l2transport
 encapsulation dot1q 10
  macsec-service decrypt-port <intf>17.10 psk-keychain <keychain_name> [policy <macsec_policy>]
```

Step 4 Configure service port.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface  <interface>17.10  l2transport
 encapsulation dot1q 10
```

Step 5 Configure the Xconnect group between ports.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# l2vpn
xconnect group local_macsec
```

```
  p2p local_macsec
   interface  <interface>15.10
   interface  <interface>16.10
```

Step 6 Connect the ports.

Example:

```
RP/0/RSP0/CPU0:router(config-if)l2vpn
 xconnect group ext_macsec
  p2p ext_macsec
   interface  <interface>17.10
   neighbor  ipv4  <a.b.c.d>  pw-id  <num>
!
```

Configuring MACsec Service for L3VPN Network

Configuring the MACsec service for L3VPN network, involves the following steps:

SUMMARY STEPS

1. Enter global configuration mode.
2. Enter interface configuration mode and configure port facing the CE router
3. Configure the PE1 router with virtual routing details.
4. Enable MACsec service.
5. Configure service port.
6. Configure the Xconnect between ports.
7. Configure ports.
8. Configure OSPF on the core interface.
9. Configure MPLS on the core interface.

DETAILED STEPS

Procedure

Step 1 Enter global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Step 2 Enter interface configuration mode and configure port facing the CE router

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/4/0/0.1
ipv4 address 161.1.1.1 255.255.255.0
encapsulation dot1q 1
```

Step 3 Configure the PE1 router with virtual routing details.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/1.1
 vrf vrf_1
 ipv4 address 161.1.1.2 255.255.255.0
 encapsulation dot1q 1
```

Step 4 Enable MACsec service.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# interface TenGigE0/3/0/0/2.1
 vrf vrf_1
 ipv4 address 181.1.1.1 255.255.255.0
 macsec-service decrypt-port TenGigE0/3/0/0/3.1 psk-keychain script_key_chain1
 encapsulation dot1q 1
```

Step 5 Configure service port.

Example:

```
RP/0/RSP0/CPU0:router(config-if)#interface TenGigE0/3/0/0/3.1 l2transport
 encapsulation dot1q 1
!
```

Step 6 Configure the Xconnect between ports.

Example:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#l2vpn
 xconnect group l3serv_xc_gp_1
  p2p l3serv_xc_p2p_1
   interface TenGigE0/3/0/0/3.1
   neighbor ipv4 3.3.3.3 pw-id 1
   !
  !
```

Step 7 Configure ports.

Example:

```
RP/0/RSP0/CPU0:router#(config)
router bgp 100
 bgp router-id 2.2.2.2
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 3.3.3.3
  remote-as 100
  update-source Loopback1
  address-family vpnv4 unicast
  !
 !
 vrf vrf_1
  rd 1234:1
  address-family ipv4 unicast
```

```
   redistribute connected
   redistribute static
  !
  neighbor 181.1.1.2
   remote-as 100
   address-family ipv4 unicast
   !
  !
 !
```

Step 8 Configure OSPF on the core interface.

Example:

```
RP/0/RSP0/CPU0:router#
macsec-PE1#sh run router ospf
router ospf core
 router-id 2.2.2.2
 redistribute connected
 redistribute static
 area 0
  interface Loopback1
  !
  interface TenGigE0/1/0/1
  !
 !
```

Step 9 Configure MPLS on the core interface.

Example:

```
RP/0/RSP0/CPU0:router#
mpls ldp
 graceful-restart
 router-id 2.2.2.2
 interface TenGigE0/1/0/1
 !
!
```

Applying MACsec Service Configuration on an Interface

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

SUMMARY STEPS

1. Enter the global configuration mode.
2. Enter the interface configuration mode.
3. If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.
4. Apply the MACsec service configuration on an interface.
5. Commit your configuration.

DETAILED STEPS

Procedure

Step 1 Enter the global configuration mode.

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Step 2 Enter the interface configuration mode.

The interface can be a physical interface or a VLAN sub-interface.

Example:

```
RP/0/RSP0/CPU0:router(config)# interface Te0/3/0/1/4
```

Step 3 If you are configuring VLAN sub-interfaces, configure the encapsulation as shown.

Example:

```
! For 802.1q encapsulation with a single tag
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 5

! For 802.1q encapsulation with double tags
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 3 second-dot1q 4

! For 802.1ad encapsulation with a single tag
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 5

! For 802.1ad encapsulation with double tags
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1ad 3 dot1ad 4
```

Step 4 Apply the MACsec service configuration on an interface.

To apply MACsec service configuration on an interface, use the following configuration.

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec-service decrypt-port TenGigE0/3/0/1/5 psk-keychain
script_key_chain1 policy mk_xpn_1tag
RP/0/RSP0/CPU0:router(config-if)# exit
```

Step 5 Commit your configuration.

Example:

```
RP/0/RSP0/CPU0:router(config)# commit
```

Verifying MACsec Encryption on IOS XR

MACsec encryption on IOS XR can be verified by running relevant commands in the Privileged Executive Mode. The verification steps are the same for MACsec encryption on L2VPN or L3VPN network.

Note With the introduction of active fallback functionality in Cisco IOS XR Software Release 7.1.2 (Release 6.7.2 for 32-bit Cisco IOS XR platforms), 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

Procedure

Step 1 Verify the MACsec policy configuration.

Example:

```
RP/0/RSP0/CPU0:router#show macsec policy mac_policy

================================================================================
Policy      Cipher         Key-Svr     Window  Conf
name        Suite          Priority    Size    Offset
================================================================================
mac_policy GCM-AES-XPN-256  0          64      30
```

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), P2MP interface (VPLS network), or VLAN sub-interface (EoMPLS PW network).

Example:

Before the introduction of active fallback functionality:

```
RP/0/RSP0/CPU0:router#show macsec mka summary

NODE: node0_0_CPU0
```

```
=================================================================
Interface     Status    Cipher Suite      KeyChain
=================================================================

Fo0/0/0/1/0   Secured   GCM-AES-XPN-256   mac_chain

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

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0
=============================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server
=============================================================================
   Fo0/0/0/1/0        d46d.5023.3709/0001     1     Secured      YES


! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/1.8
=============================================================================
      Interface         Local-TxSCI       # Peers   Status  Key-Server
=============================================================================
  Fo0/0/0/1/1.8      e0ac.f172.4124/001d      1     Secured     Yes
```

With the introduction of active fallback functionality:

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

```
RP/0/RSP0/CPU0:router#show macsec mka summary

NODE: node0_0_CPU0
=============================================================================================
   Interface-Name     Status     Cipher-Suite       KeyChain       PSK/EAP      CKN
=============================================================================================
     Fo0/0/0/1/0      Secured  GCM-AES-XPN-256      mac_chain      PRIMARY      5555
     Fo0/0/0/1/0      Active   GCM-AES-XPN-256      mac_chain_fb   FALLBACK     5556

Total MACSec Sessions : 2
     Secured Sessions : 1
     Pending Sessions : 0
      Active Sessions : 1

RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0
=============================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
=============================================================================================
   Fo0/0/0/1/0        d46d.5023.3709/0001     1      Secured      YES      PRIMARY     5555
   Fo0/0/0/1/0        d46d.5023.3709/0001     1      Active       YES     FALLBACK     5556
```

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 **Init**.

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

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

Example:

```
RP/0/RSP0/CPU0:router#show macsec mka session

NODE: node0_0_CPU0

================================================================
Interface    Local-TxSCI           # Peers   Status  Key-Server
================================================================

Fo0/0/0/1/0  001d.e5e9.aa39/0005      1      Secured  YES
```

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_1_CPU0
=============================================================================================================
   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
=============================================================================================================
     Fo0/0/0/1/0      001d.e5e9.aa39/0005     1      Secured      NO       PRIMARY     1234
     Fo0/0/0/1/0      001d.e5e9.aa39/0005     1      Active       NO       FALLBACK    1111
```

The **#Peers** field in the output confirms the presence of the peer you have configured on the physical interface, **Fo0/0/0/1/0**. 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.

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.

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
Fri May 28 07:18:45.726 UTC

NODE: node0_0_CPU0
=============================================================================================
Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
=============================================================================================
  Te0/0/0/1       6c8b.d34f.0635/0001     2      Secured      NO      FALLBACK     5556
```

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

Example:

Before the introduction of active fallback functionality:

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Fo0/0/0/1/0 detail
MKA Detailed Status for MKA Session
```

```
==================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : 001d.e5e9.aa39/0005
Local Tx-SSCI           : 1
Interface MAC Address   : 001d.e5e9.aa39
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0
CAK Name (CKN)          : 1020000000000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A880BB45B9CE01584535F239
Message Number (MN)     : 5382
Authenticator           : NO
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : A880BB45B9CE01584535F23900000001 (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)
MKA Policy Name         : scale-21
Key Server Priority     : 20
Replay Window Size      : 40
Confidentiality Offset  : 50
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000001 (GCM-AES-128)
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
  -----------------------------------------------------------------------------
  4E33A276E7F79C04D80FE346       27114  d46d.5023.3704/0001     2         235
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  -----------------------------------------------------------------------------
```

! If sub-interfaces are configured, the output would be as follows:

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Fo0/0/0/1/1.8 detail
MKA Detailed Status for MKA Session
==================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4124/001d
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4124
MKA Port Identifier     : 29
Interface Name          : Fo0/0/0/1/1.8
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : 1EC4A4D1B0D75D3D5C2F6393
Message Number (MN)     : 1915
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 3
Latest SAK KI (KN)      : EB1E04894327E4EFA283C66200000003 (3)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (4)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
```

```
MKA Policy Name         : test12
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
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 (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  EB1E04894327E4EFA283C662        1908  001d.e5e9.b1c0/0037     2           0
Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
RP/0/RSP0/CPU0:macsec-CE1#sh macsec mka  interface Fo0/0/0/1/1.8
============================================================
Interface-name          KeyChain-Name           Policy Name
============================================================
Fo0/0/0/1/1.8           kc3                     test12


! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface FortyGigE0/0/0/1/0.1 detail
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI            : e0ac.f172.4123/0001
Local Tx-SSCI           : 1
Interface MAC Address   : e0ac.f172.4123
MKA Port Identifier     : 1
Interface Name          : Fo0/0/0/1/0.1
CAK Name (CKN)          : ABC1000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : A1DB3E42B4A543FBDBC281A6
Message Number (MN)     : 1589
Authenticator           : NO
Key Server              : NO
MKA Cipher Suite        : AES-128-CMAC
Latest SAK Status       : Rx & Tx
Latest SAK AN           : 1
Latest SAK KI (KN)      : AEC899297F5B0BDEF7C9FC6700000002 (2)
Old SAK Status          : No Rx, No Tx
Old SAK AN              : 0
Old SAK KI (KN)         : RETIRED (1)
SAK Transmit Wait Time  : 0s (Not waiting for any peers to respond)
SAK Retire Time         : 0s (No Old SAK to retire)
MKA Policy Name         : mk_xpn1
Key Server Priority     : 0
Replay Window Size      : 1024
Confidentiality Offset  : 50
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           : 2
# of MACsec Capable Live Peers Responded : 0
Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ----------------------------------------------------------------------------
  AEC899297F5B0BDEF7C9FC67         225  001d.e5e9.b1bf/0001     3           0
  0A4C49EE5B7401F1BECB7E22         147  001d.e5e9.f329/0001     2           0
```

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

With the introduction of active fallback functionality:

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

```
RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/0/11 detail

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

Local Tx-SCI                    : 7061.7bea.1df4/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1df4
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/0/11
CAK Name (CKN)                  : 2111
CA Authentication Mode          : PRIMARY-PSK
Keychain                        : test1
Member Identifier (MI)          : 42A78BD6243539E917B8C6B2
Message Number (MN)             : 555
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)              : 69B39E87B3CBA673401E989100000001 (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                 : P12
Key Server Priority             : 20
Delay Protection                : TRUE
Replay Window Size              : 100
Include ICV Indicator           : TRUE
Confidentiality Offset          : 0
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
------------------------------------------------------------------------------
69B39E87B3CBA673401E9891      617       008a.96d6.194c/0001    2        20

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

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

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:27:56.548
Peer Count             : 1

RxSCI                  : 008A96D6194C0001
  MI                    : 69B39E87B3CBA673401E9891
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 18 13:27:56.518

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

Local Tx-SCI                    : 7061.7bea.1df4/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1df4
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/0/11
CAK Name (CKN)                  : 2000
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : test1f
Member Identifier (MI)          : 1BB9428C721F6EE3E538C942
Message Number (MN)             : 553
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)              : 69B39E87B3CBA673401E989100000001 (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                 : P12
Key Server Priority             : 20
Delay Protection                : TRUE
Replay Window Size              : 100
Include ICV Indicator           : TRUE
Confidentiality Offset          : 0
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
--------------------------------------------------------------------------------
8F59AD6021FA3E2D5F9E6231      615       008a.96d6.194c/0001    2        20

Potential Peer List:
--------------------------------------------------------------------------------
```

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

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:27:56.547
Peer Count             : 1

RxSCI                  : 008A96D6194C0001
  MI                   : 8F59AD6021FA3E2D5F9E6231
  Peer CAK             : Match
  Latest Rx MKPDU      : 2021 May 18 13:27:56.518

RP/0/RSP0/CPU0:router#
```

If sub-interfaces are configured, the output would be as follows. In this example, the status of FALLBACK-PSK is *Secured*.

```
RP/0/RSP0/CPU0:router# show macsec mka session interface Hu0/0/0/0.6 detail
MKA Detailed Status for MKA Session
==================================
Status: Secured - Secured MKA Session with MACsec

Local Tx-SCI                    : 7061.7bea.1dc8/0006
Local Tx-SSCI                   : 1
Interface MAC Address           : 7061.7bea.1dc8
MKA Port Identifier             : 6
Interface Name                  : Hu0/0/0/0.6
CAK Name (CKN)                  : 9999
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : D_tagf
Member Identifier (MI)          : 1DE18714A098B80964CC651E
Message Number (MN)             : 6203
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                   : 0
Latest SAK KI (KN)              : 1DE18714A098B80964CC651E00000001 (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               : 23510s
Time to exit suspension         : NA

MKA Policy Name                 : D_tag1
Key Server Priority             : 1
Delay Protection                : FALSE
Replay Window Size              : 1000
Include ICV Indicator           : TRUE
Confidentiality Offset          : 50
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
```

```
# of MACSec Suspended Peers              : 0

Live Peer List:
----------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
----------------------------------------------------------------------------
5C852D8F920306893D2BFB8F     10978      00c1.645f.2dd4/0006    2        11

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

Suspended Peer List:
----------------------------------------------------------------------------
       Rx-SCI          SSCI
----------------------------------------------------------------------------

Peers Status:
Last Tx MKPDU          : 2021 May 18 13:29:15.687
Peer Count             : 1

RxSCI                  : 00C1645F2DD40006
  MI                    : 5C852D8F920306893D2BFB8F
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 18 13:29:15.769

RP/0/RSP0/CPU0:router#


! In a VPLS network with multipoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/1/7 detail
Fri May 28 07:19:11.362 UTC

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

Local Tx-SCI                   : 6c8b.d34f.0635/0001
Local Tx-SSCI                  : 2
Interface MAC Address          : 6c8b.d34f.0635
MKA Port Identifier            : 1
Interface Name                 : Te0/0/0/1
CAK Name (CKN)                 : 5556
CA Authentication Mode         : FALLBACK-PSK
Keychain                       : test2f
Member Identifier (MI)         : 6D14ECCDFB70E7E0463BD509
Message Number (MN)            : 20455
Authenticator                  : NO
Key Server                     : NO
MKA Cipher Suite               : AES-256-CMAC
Configured MACSec Cipher Suite : GCM-AES-XPN-256

Latest SAK Status              : Rx & Tx
Latest SAK AN                  : 2
Latest SAK KI (KN)             : 1BBDDC0520C797C26AB7F1BF00000002 (2)
Old SAK Status                 : No Rx, No Tx
Old SAK AN                     : 1
Old SAK KI (KN)                : RETIRED (1)

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           : 2
# of MACsec Capable Live Peers Responded : 0

Live Peer List:
------------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
------------------------------------------------------------------------------
1BBDDC0520C797C26AB7F1BF     19997      008a.96d6.194c/0001    3        16
B25B1000CC6FAE92D1F85738      139       dc77.4c3e.59c3/0001    1        16

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

Peers Status:
 Last Tx MKPDU          : 2021 May 28 07:19:10.153
 Peer Count             : 2

 RxSCI                  : 008A96D6194C0001
  MI                    : 1BBDDC0520C797C26AB7F1BF
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 07:19:09.960

 RxSCI                  : DC774C3E59C30001
  MI                    : B25B1000CC6FAE92D1F85738
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 07:19:10.180

RP/0/RSP0/CPU0:router#


RP/0/RSP0/CPU0:router#show macsec mka session interface Hu0/0/1/7.1 detail

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

Local Tx-SCI                    : 7061.7bff.e5e8/0001
Local Tx-SSCI                   : 2
Interface MAC Address           : 7061.7bff.e5e8
MKA Port Identifier             : 1
Interface Name                  : Hu0/0/1/7.1
CAK Name (CKN)                  : 5556
CA Authentication Mode          : FALLBACK-PSK
Keychain                        : test22f
Member Identifier (MI)          : 8FF3D1BBF09EA4AD6A0FC1B5
Message Number (MN)             : 81
Authenticator                   : NO
```

```
Key Server                      : YES
MKA Cipher Suite                : AES-256-CMAC
Configured MACSec Cipher Suite  : GCM-AES-XPN-256

Latest SAK Status               : Rx & Tx
Latest SAK AN                   : 3
Latest SAK KI (KN)              : 8FF3D1BBF09EA4AD6A0FC1B500000002 (2)
Old SAK Status                  : No Rx, No Tx
Old SAK AN                      : 2
Old SAK KI (KN)                 : RETIRED (1)

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               : 17930s
Time to exit suspension         : NA

MKA Policy Name                 : P123
Key Server Priority             : 10
Delay Protection                : FALSE
Replay Window Size              : 64
Include ICV Indicator           : FALSE
Confidentiality Offset          : 30
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           : 2
# of MACsec Capable Live Peers Responded : 2

# of MACSec Suspended Peers              : 0

Live Peer List:
--------------------------------------------------------------------------------
           MI                  MN             Rx-SCI          SSCI  KS-Priority
--------------------------------------------------------------------------------
6BCF91135F807CB9F57DDAAA       61       dc77.4c3e.5b05/0001    1        24
D81CFE93D07E932DDC33666E       44       00a7.4250.56c2/0001    3        25

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

Suspended Peer List:
--------------------------------------------------------------------------------
       Rx-SCI          SSCI
--------------------------------------------------------------------------------

Peers Status:
 Last Tx MKPDU          : 2021 May 28 13:16:50.992
 Peer Count             : 2

 RxSCI                  : DC774C3E5B050001
  MI                    : 6BCF91135F807CB9F57DDAAA
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 13:16:51.312

 RxSCI                  : 00A7425056C20001
  MI                    : D81CFE93D07E932DDC33666E
  Peer CAK              : Match
  Latest Rx MKPDU       : 2021 May 28 13:16:50.945
RP/0/RSP0/CPU0:router#
```

Step 5 Verify the MACsec session counter statistics.

Example:

```
RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/0


MKA Statistics for Session on interface (Fo0/0/0/1/0)
=====================================================
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.......... 3
SAKs Rekeyed............ 2
SAKs Received........... 0
SAK Responses Received.. 3

MKPDU Statistics
MKPDUs Transmitted...... 5425
"Distributed SAK".. 8
"Distributed CAK".. 0
MKPDUs Validated & Rx... 4932
"Distributed SAK".. 0
"Distributed CAK".. 0

MKA IDB Statistics
MKPDUs Tx Success.......... 5425
MKPDUs Tx Fail............. 0
MKPDUS Tx Pkt build fail... 0
MKPDUs Rx CA Not found..... 0
MKPDUs Rx Error............ 0
MKPDUs Rx Success.......... 4932

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

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

! If sub-interfaces are configured, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka statistics interface Fo0/0/0/1/1.8

MKA Statistics for Session on interface (Fo0/0/0/1/1.8)
=======================================================
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........... 9
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1973
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 1965
      "Distributed SAK".. 9
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1973
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1965
```

```
! In a VPLS network with a mulitpoint interface, the output would be as follows:

RP/0/RSP0/CPU0:router# show macsec mka statistics interface FortyGigE0/0/0/1/0.1

MKA Statistics for Session on interface (Fo0/0/0/1/0.1)
=======================================================
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........... 2
   SAK Responses Received.. 0
MKPDU Statistics
   MKPDUs Transmitted...... 1608
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 406
      "Distributed SAK".. 2
      "Distributed CAK".. 0
MKA IDB Statistics
   MKPDUs Tx Success.......... 1608
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 1802
```

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 ASR 9000

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. Use the IDB handle retrieved from Step 1 to verify the platform hardware information.
3. Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.
4. Verify the MACsec Secure Channel (SC) information programmed in the hardware.

DETAILED STEPS

Procedure

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

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface Fo0/0/0/1/0


IDB Details:
if_sname : Fo0/0/0/1/0
if_handle : 0x3480
Replay window size : 64
Local MAC : 00:1d:e5:e9:aa:39
Rx SC Option(s) : Validate-Frames Replay-Protect
Tx SC Option(s) : Protect-Frames Always-Include-SCI
Security Policy : MUST SECURE
Sectag offset : 8
VLAN : Outer tag (etype=0x8100, id=1, priority=0, cfi=0): Inner tag (etype=0x8100, id=1, priority=0,
 cfi=0)
Rx SC 1
Rx SCI : 001de5e9b1bf0019
Peer MAC : 00:1d:e5:e9:b1:bf
Stale : NO
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ad 7b 6f 63 16 09 8f f3 d2
Rx SA Program Req[0]: 2015 Oct 09 15:20:53.082
Rx SA Program Rsp[0]: 2015 Oct 09 15:20:53.092

Tx SC
Tx SCI : 001de5e9aa39001a
Active AN : 0
Old AN : 255
Next PN : 1, 0, 0, 0
```

```
SAK Data
SAK[0] : ***
SAK Len : 32
HashKey[0] : ***
HashKey Len : 16
Conf offset : 30
Cipher Suite : GCM-AES-XPN-256
CtxSalt[0] : 83 c3 7b ae 7b 6f 63 16 09 8f f3 d2
Tx SA Program Req[0]: 2015 Oct 09 15:20:55.053
Tx SA Program Rsp[0]: 2015 Oct 09 15:20:55.064
```

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

```
RP/0/RSP0/CPU0:router# show macsec ea idb interface FortyGigE0/0/0/1/0.1
IDB Details:
  if_sname                : Fo0/0/0/1/0.1
  if_handle               : 0x2e40
  Replay window size      : 1024
  Local MAC               : e0:ac:f1:72:41:23
  Rx SC Option(s)         : Validate-Frames Replay-Protect
  Tx SC Option(s)         : Protect-Frames Always-Include-SCI
  Security Policy         : MUST SECURE
  Sectag offset           : 8
  VLAN                    : Outer tag (etype=0x8100, id=1, priority=0, cfi=0)
                          : Inner tag (etype=0x8100, id=1, priority=0, cfi=0)
  Rx SC 1
    Rx SCI                : 001de5e9f3290001
    Peer MAC              : 00:1d:e5:e9:f3:29
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[1]          : ae ca 99 2b 7f 5b 0b de f7 c9 fc 67
  Rx SC 2
    Rx SCI                : 001de5e9b1bf0001
    Peer MAC              : 00:1d:e5:e9:b1:bf
    Stale                 : NO
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[1]          : ae ca 99 2a 7f 5b 0b de f7 c9 fc 67
  Tx SC
    Tx SCI                : e0acf17241230001
    Active AN             : 1
    Old AN                : 0
    Next PN               : 1, 1, 0, 0
    SAK Data
      SAK[1]              : ***

      SAK Len             : 32
      HashKey[1]          : ***
      HashKey Len         : 16
      Conf offset         : 50
```

```
Cipher Suite        : GCM-AES-XPN-256
CtxSalt[1]          : ae ca 99 28 7f 5b 0b de f7 c9 fc 67
```

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 Use the IDB handle retrieved from Step 1 to verify the platform hardware information.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware
idb location 0/0/CPU0 | b 3480

if_handle : 0x00003480
NPPort : 099 [0x063]
LdaPort : 016 [0x010] SerdesPort : 000 [0x000]
NetSoftPort : 061 [0x03d] SysSoftPort : 062 [0x03e]
Active AN : 0x00000000 Idle AN : 0x000000ff
Match-All Tx SA : 0x80010001 Match-All Rx SA : 0x00010001
Match-All Tx Flow : 0x80000003 Match-All Rx Flow : 0x00000003
Bypass Tx SA : 0x80000000 Bypass Rx SA : 0x00000000
Tx SA[0] : 0x80020002 Tx Flow[0] : 0x8000000c
Tx SA[1] : 0xffffffff Tx Flow[1] : 0xffffffff
Tx SA[2] : 0xffffffff Tx Flow[2] : 0xffffffff
Tx SA[3] : 0xffffffff Tx Flow[3] : 0xffffffff
Rx SA[0] : 0x00020002 Rx Flow[0] : 0x0000000c
Rx SA[1] : 0xffffffff Rx Flow[1] : 0xffffffff
Rx SA[2] : 0xffffffff Rx Flow[2] : 0xffffffff
Rx SA[3] : 0xffffffff Rx Flow[3] : 0xffffffff
```

Step 3 Use the Transmitter SA retrieved from Step 2 to verify the MACsec SA information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware sa
0x80020002 interface Fo0/0/0/1/0 location 0/0/CPU0

MACSEC HW SA Details:
Action Type : 0x00000003
Direction : Egress
Dest Port : 0x00000000
Conf Offset : 00000030
Drop Type : 0x00000002
Drop NonResvd : 0x00000000
SA In Use : YES
ConfProtect : YES
IncludeSCI : YES
ProtectFrame : YES
UseEs : NO
UseSCB : NO
SCI : 00 1d e5 e9 aa 39 00 05
Replay Window : 64 MacsecCryptoAlgo : 7
Direction : Egress AN : 0
```

```
AES Key Len : 256 X-Packet Number : 0x0000000000000000
CtxSalt : f8d88dc3e1c5e6a94ca2299
```

The output displays the details of the encryption, such as the AES key, the Auth key, and other parameters.

Step 4 Verify the MACsec Secure Channel (SC) information programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router# show macsec ea platform hardware msc
interface Fo0/0/0/1/0 location 0/0/CPU0


MACSEC HW Cfg Details:
Mode : 0x5
Counter Clear on Read : 0x0
SA Fail Mask : 0xffff
VlanCounter Update : 0x1
Global SecFail Mask : 0xffffffff
Latency : 0xff
StaticBypass : 0x0
Should secure : 0x0
Global Frame Validation : 0x2
Ctrl Pkt CC Bypass : 0x1
NonCtrl Pkt CC Bypass : 0x1
Sequence Number Threshold : 0xbffffb8
Sequence Number Threshold 64bit : 0x000002fffffffffd
Non Matching Non Control Pkts Programming
      Untagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
Non Matching Control Pkts Programming
      Untagged : Bypass: 0x1 DestPort : 0x2, DropType : 0xffffffff
      Tagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      BadTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
      KayTagged : Bypass: 0x0 DestPort : 0x2, DropType : 0x2
```

This completes the verification of MACsec encryption on the router hardware.

This completes the configuration and verification of MACsec encryption.

Quantum safe key distribution options for MACsec

Quantum computers are a threat to existing cryptographic algorithms. To address this problem, you can use session keys to establish a secure connection between two routers.

Cisco offers two solutions to derive session keys:

- Session Key Service (SKS): Used to derive the session keys on both the routers establishing the MACsec connection without using an external key source.
- Secure Key Integration Protocol (SKIP): Used to derive the session keys on both the routers establishing the MACsec connection using an external server. Enables a router to securely import a post-quantum pre-shared key (PPK) from an external key source such as a quantum key distribution (QKD) device.

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Session key service	Release 7.9.1	The router integrates the Session Key Service (SKS) as a software component, allowing it to generate and manage the cryptographic keys needed for quantum-safe MACsec. By using SKS, you can implement MACsec without requiring additional hardware, simplifying deployment and reducing costs. The SKS software should be present on the peer routers. For more information on Quantum Key Distribution, see Post Quantum Security Brief.

Session key service

The Session Key Service (SKS) is a cryptographic service that manages symmetric keys for encryption and decryption. These are the salient features of SKS on a router.

Key generation

The SKS engine is a software component within the router responsible for generating cryptographic keys. It creates keys that will be used to encrypt and decrypt data between peer routers.

No additional hardware required

This implies that the key generation and exchange process does not need extra hardware components. The SKS engine functions with the existing router hardware, making it cost-effective and easy to deploy.

Seed protected by McEliece cryptosystem

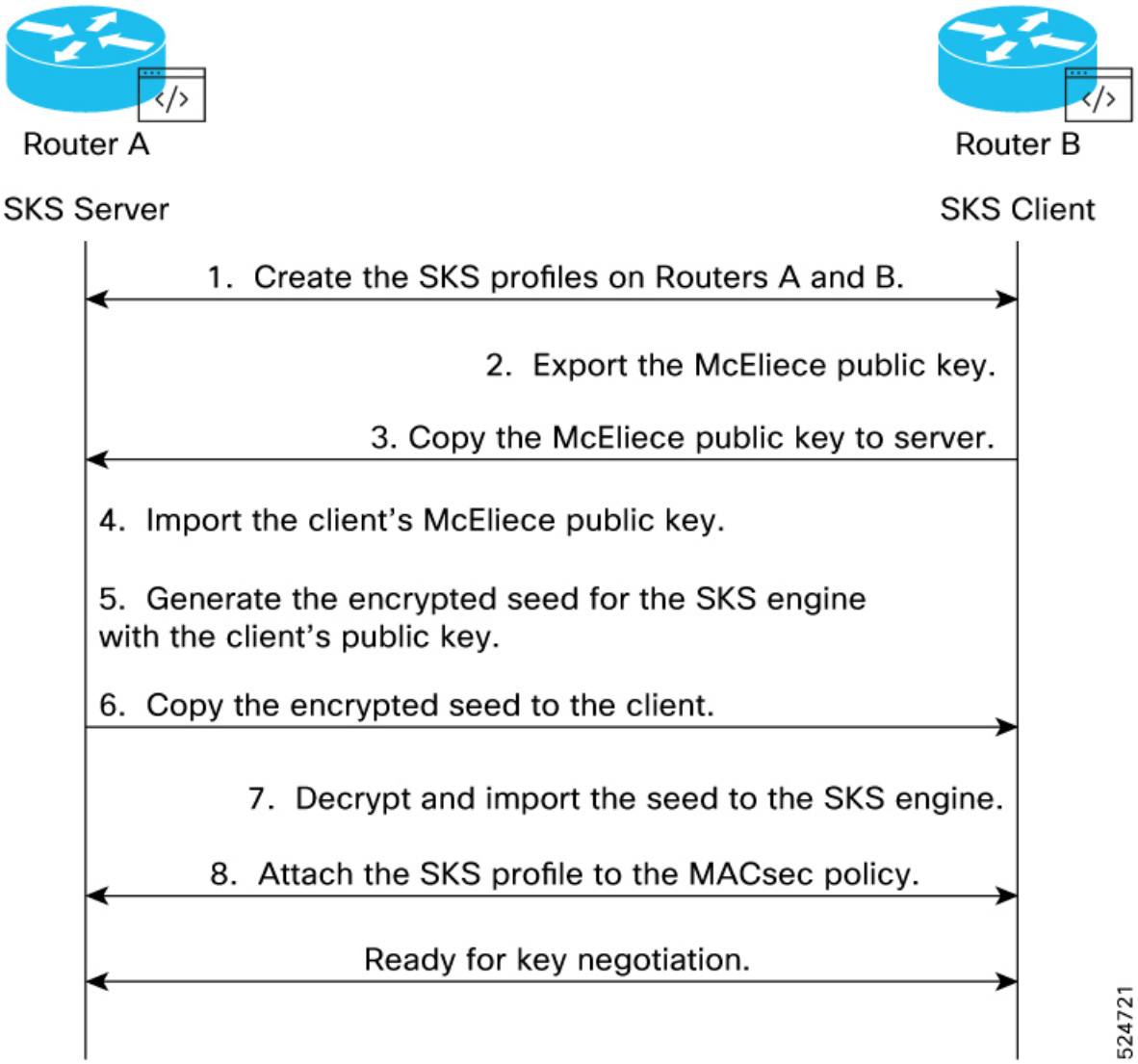
Seeding refers to initializing the SKS with a specific value, called a seed, which ensures that both communicating peers generate the same cryptographic keys. This is crucial for successful encryption and decryption. The McEliece cryptosystem is a public-key cryptosystem known for its resistance to quantum computer attacks. Protecting the seed with McEliece ensures it remains secure against future quantum computing threats.

Only Key ID sent on the network

Instead of sending the entire cryptographic key over the network, the router sends only a Key Identifier (Key ID). The receiving peer uses the Key ID to derive the corresponding key locally using its SKS. Using the Key ID enhances security by minimizing exposure of the actual key during transmission.

Configure SKS

The SKS software component on a router is used to configure the server and client to generate preshared keys for quantum-safe MACsec. This image depicts the steps you should perform to generate preshared keys. You can click a step to learn more.



1 Create the SKS profiles on Routers A and B.
2 Export the McEliece public key.
3 Copy the McEliece public key to the server.
4 Import the client's McEliece public key.
5 Generate the encrypted seed for the SKS engine with the clients public key.
6 Copy the encrypted seed to the client.
7 Decrypt and import the seed to the sks engine.
8 Attach the SKS profile to the MACsec policy.
9 Ready for key negotiation.

Create the SKS profile on the server and client

Follow these steps to create the SKS profile on the server and client.

Before you begin

First, let us gather the required details to facilitate the devices to agree on encryption parameters to establish a secure connection:

- Routers A and B are equipped with the SKS engine.
- Router A acts as the server and Router B acts as the client.
- Routers A and B are ready for key negotiation after all the steps are performed and the SKS profile is attached to the MACsec policy.

Procedure

Step 1 Enter the **sks profile** <*profile-name*> **device-identifier** <*name of peer*> command to apply a profile to manage secure communications.

Server configuration

```
Router A(config)# sks profile prof-A device-identifier peer-1
```

Client configuration

```
Router B(config)#sks profile prof-B device-identifier peer-2
```

Step 2 Enter the **live-key** <*number of MACsec sessions*> command to manage the active keys used in MACsec sessions.

Server configuration

```
Router A(config-sks-profile)#live-keys 5
```

Client configuration

```
Router B(config-sks-profile)#live-keys 5
```

Step 3 Enter the **peer identifier** <*peer-name*> command to associate the server with the client and the client with the server.

Server configuration

```
Router A(config-sks-profile)#peer-identifier peer-2
```

Client configuration

```
Router B(config-sks-profile)#peer-identifier peer-1 master
```

Export the McEliece public key

Follow these steps to export the McEliece public key.

Procedure

Step 1 Enter the **crypto sks key export mceliece** command to export the public key to the client.

Client configuration

```
Router B#crypto sks key export mceliece
```

The message **Pubkey exported file: disk0:/MeCe_the_MC_default_pub** is displayed.

Step 2 Verify the default path where the public key is exported.

Client configuration

```
Router B#dir disk0:/MeCe_the_MC_default_pub
Mon Feb 24 04:25:25.013 UTC

Directory of disk0:/MeCe_the_MC_default_pub
73 -rw-r--r--. 1 1357824 Feb 24 04:20 MeCe_the_MC_default_pub

989244 kbytes total (919872 kbytes free)
```

Copy the McEliece public key to the server

Follow these steps to copy the McEliece public key to the server.

Before you begin

In the example, **disk0:/MeCe_the_MC_default_pub** is the source path of the client and **cisco@1.2.42.3** is the IP address of the server.

Procedure

Step 1 Using the Secure Copy Protocol (SCP), enter the **scp / disk0\:/**<*source path on the client* <*destination path of the server*>**:/disk0:/** to copy the files from the client to the server.

Client configuration

```
Router B# scp /disk0\:/MeCe_the_MC_default_pub cisco@1.2.42.3:/disk0:/
```

This command has securely copied the file named **MeCe_the_MC_default_pub** from the local directory **/disk0:/** to the remote directory **/disk0:/** on the host **1.2.42.3** using the Cisco user account.

Step 2 Verify that the files are copied from the client to the server.

Client configuration

```
Router B #dir disk0:/MeCe_the_MC_default_pub
Mon Feb 24 04:27:00.398 UTC

Directory of disk0:/MeCe_the_MC_default_pub
73 -rw-r--r--. 1 1357824 Feb 24 04:26 MeCe_the_MC_default_pub

989244 kbytes total (919868 kbytes free)
```

Import the client McEliece public key

Follow these steps to import the client McEliece public key.

Before you begin

In the example, peer 2 is Router B's name to which the key corresponds. The local directory **disk0:/MeCe_the_MC_default_pub** is the source path of the key on the server that was copied in the previous step.

Procedure

Step 1 Enter the **crypto sks key import mceliece** *<client's name> <source path of the key on the server>* command to import the McEliece public key to the server.

Server configuration

```
Router A# crypto sks key import mceliece peer-2 disk0:/MeCe_the_MC_default_pub
```

Step 2 Verify that the **Pubkey Import Done** is set to **True** for the required peer.

Server configuration

```
Router A# show crypto sks peer all
Mon Feb 24 04:29:06.492 UTC
Peer Name          : peer-2
Profile Name       : prof-A
Seed Done          : TRUE
Pubkey Import Done : TRUE
Master             : FALSE
```

Generate the encrypted seed for the SKS engine with the client public key

Follow these steps to generate the encrypted seed for the SKS engine with the client public key.

Procedure

Step 1 Enter the **crypto sks seed export mceliece** *<client name>* command to generate and export the seed to the server.

Server configuration

```
Router A# crypto sks seed export mceliece peer-2
```

The command exports a McEliece cryptographic seed that is associated with Router B.

Step 2 Verify if an encrypted seed is exported to the */disk0\:/enc_self_peer-2* location.

Server configuration

```
Router A# dir disk0:/enc_self_peer-2
Mon Feb 24 04:35:57.596 UTC

Directory of disk0:/enc_self_peer-2
74 -rw-r--r--. 1 480 Feb 24 04:35 enc_self_peer-2

989244 kbytes total (919860 kbytes free)
```

Copy the encrypted seed to the client

Follow these steps to copy the encrypted seed to the client.

Procedure

Step 1 Using the Secure Copy Protocol (SCP), enter the **scp /disk0\:/**<*source path of the client*> <*destination path of the server*> command to copy the files from the server to the client.

Server configuration

```
Router A# scp /disk0\:/enc_self_peer-2 cisco@1.2.43.3:/disk0:/
```

Step 2 Verify that the files are copied from the server to the client.

Client configuration

```
Router B# dir disk0:/enc_self_peer-2
Mon Feb 24 04:35:57.596 UTC

Directory of disk0:/enc_self_peer-2
74 -rw-r--r--. 1 480 Feb 24 04:35 enc_self_peer-2

989244 kbytes total (919860 kbytes free)
```

Decrypt and import the seed to the sks engine

Follow these steps to decrypt and import the seed to the sks engine.

Procedure

Step 1 Enter the **crypto sks seed import mceliece** <*server name*> <*server path*> command to import the seed on the client.

Client configuration

```
Router B# crypto sks seed import mceliece peer-1 disk0:/enc_self_peer-2
```

The seed is associated with Router A and **disk0:/enc_self_peer-2** is the file path from which the seed is being imported. It indicates that the seed is stored in a file located at **disk0:/enc_self_peer-2** on the router.

Step 2 Verify that the seed is imported to the client.

Client configuration

```
Router B# show crypto sks peer all
Mon Feb 24 04:37:35.578 UTC
Peer Name          : peer-1
Profile Name       : prof-B
Seed Done          : TRUE
Pubkey Import Done : FALSE
Master             : TRUE
------------------------------
```

Attach the SKS profile to the MACsec policy

Follow these steps to create the SKS profile on the server and client.

Procedure

Step 1 Configure the SKS profile on both MACsec peers.

Server configuration

```
Router A(config)#macsec-policy p1
Router A(config-macsec-policy-p1)#ppk
Router A(config-macsec-policy-p1-ppk)#sks-profile prof-A
```

Client configuration

```
Router B(config)#macsec-policy p2
Router B(config-macsec-policy-p1)#ppk
Router B(config-macsec-policy-p1-ppk)#sks-profile prof-B
```

Step 2 Verify the MACsec configuration.

Server configuration

```
Router A#show macsec mka session

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

Local Tx-SCI                    : c847.091c.d060/0001
Local Tx-SSCI                   : 1
Interface MAC Address           : c847.091c.d060
MKA Port Identifier             : 1
Interface Name                  : Te0/0/0/24
CAK Name (CKN)                  : 4000
CA Authentication Mode          : PRIMARY-PSK
Keychain                        : tet
Member Identifier (MI)          : 2E222A17F6E9535E1ACD4747
Message Number (MN)             : 333807
Authenticator                   : NO
Key Server                      : NO
MKA Cipher Suite                : AES-256-CMAC
Configured MACSec Cipher Suite  : GCM-AES-XPN-256
Key Distribution Mode           : PPK
------------<truncated>--------------
```

Ready for key negotiation: Once the routers attach the SKS profile to the MACsec policy, they have met all technical and security prerequisites. Routers A and B can now establish a secure communication link using symmetric key cryptography.

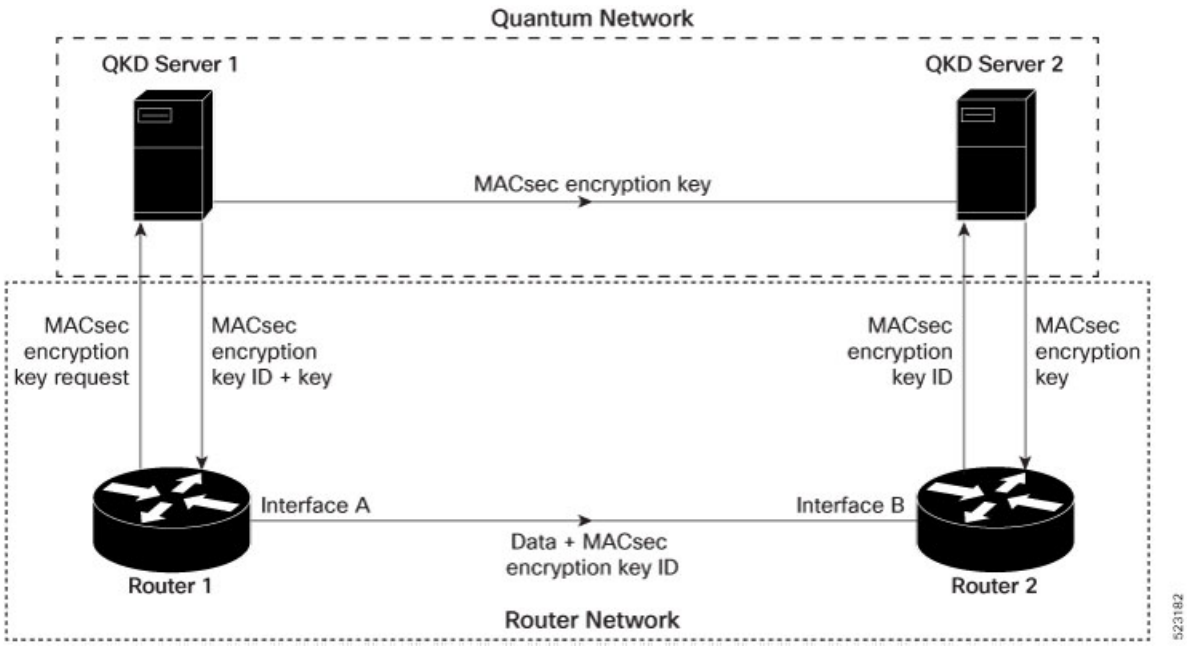
Understanding SKIP

Table 11: Feature History Table

Feature Name	Release Information	Feature Description

Secure Key Integration Protocol for Cisco IOS XR Routers	Release 7.10.1	We have now enabled Secure Key Integration Protocol (SKIP), a key-exchange protocol, on your routers to ensure a long-term secure MACsec. This is made possible because the SKIP protocol facilitates communication with external quantum devices, thereby enabling your routers to use Quantum Key Distribution (QKD) to create and transmit secure MACsec keys. Using QKD overcomes a critical 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 these changes:
		• **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)

Cisco Secure Key Integration Protocol (SKIP) enables your router that supports encryption to use keys by a quantum distribution system. SKIP implementation in Cisco IOS XR software supports integrating external Quantum Key Distribution (QKD) devices with your routers. With integration support between the routers and QKD devices, you can use the QKD devices to exchange encryption keys for communication between the routers. And this mechanism eliminates the key distribution problem in a post quantum world.



Quantum Key Distribution (QKD) is a method for securely transmitting a secret key between two parties. QKD uses the laws of quantum mechanics to guarantee security even when eavesdroppers monitor the communication channel. In QKD, the key is encoded in the states of single photons. The QKD transmits the keys over optical fiber or free space (vacuum). The security of the key relies on the fact that measuring a quantum state introduces a change in the quantum state. The change in quantum states helps the two end parties of the communication channel to identify any interception of their key.

QKD is a secure key exchange mechanism against quantum attacks and will remain so, even with future advancements in cryptanalysis or quantum computing. Unlike other cryptographic algorithms, QKD doesn't need continual updates based on discovered vulnerabilities.

Feature Highlights

- You can use the QKD devices in the following combinations:
  - Same QKD device on the end ports of the peer routers
  - Different QKD devices on the end ports of the peer routers
  - Multiple links between the same peer routers using different QKD devices
- You can use a specific source interface for the router communication with the QKD devices. To use a specific source interface, configure the source interface in the QKD profile. Use the **source interface** command in SKS configuration mode as follows.

```
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
```

- You can use an HTTP Proxy for the router communication with the QKD devices. Use the following configuration for the router to use an HTTP proxy server to communicate to the QKD devices.

```
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
```

Note The **http proxy server** command supports configuration using IPv4 address, IPv6 address, and hostname of the HTTP proxy.

Restrictions

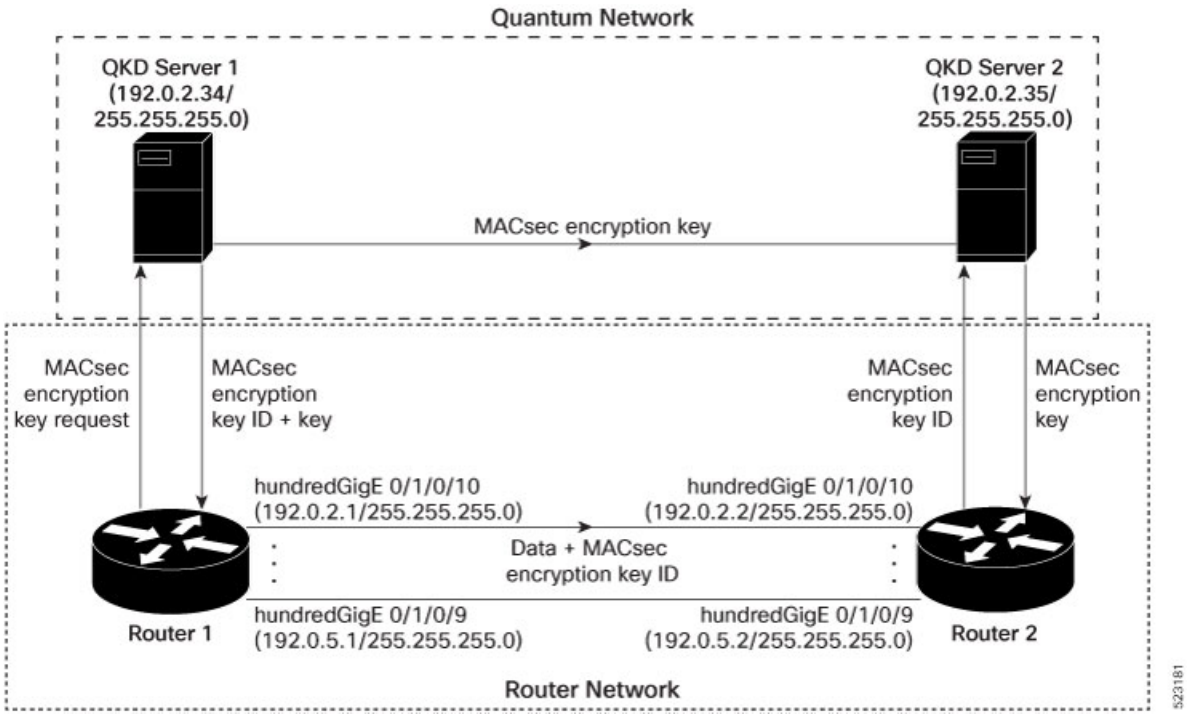
The following section lists the restriction to consider prior implementing SKIP:

- You can use the SKIP protocol only in a Point to Point MACsec link encryption scenario.
- The SKIP protocol is available only on the interfaces that support MACsec encryption.

Configure Point-to-Point MACsec Link Encryption using SKIP

In Point-to-Point MACsec Link Encryption, the router uses SKIP to establish secure encryption. This encryption is set up between two interfaces in peer routers and requires the assistance of an external QKD device network. The QKD network shares the MACsec encryption key instead of the router network. Thus, when the router needs to create a MACsec link between peer router interfaces, it contacts the external QKD device and requests the key. The external QKD device generates a Key pair comprising the Key ID and the Key. The Key ID serves as the unique identification string for the Key (Shared Secret). The QKD then shares both the Key ID and Key with the router and the router shares only the Key ID with its peer. The Peer router uses this Key ID to retrieve encryption keys from its QKD device. Therefore, Quantum networks securely communicate encryption keys always.

Figure 10: Point-to -Point MACsec Link Encryption using SKIP



Prerequisites

- Configure MACsec Pre-Sared Key (PSK). For more information, see MACsec PSK, on page 9.
- Configure MACsec in the PPK mode.
- An external QKD devices network.
- To establish secured connection (https) from the router to the QKD server, you must import the same root CA which signed the QKD server also. For this, you must:
  - Add the QKD server CA to the trustpoint in the router. For more information, see Configure Trustpoint.
  - Import the QKD server root CA certificate in the router. For more information, see Configure Certificate Enrollment Using Cut-and-Paste.

Configuration

The following example details how to establish Point to Point MACsec Link Encryption using SKIP:

Router 1:

1. Configure the QKD profile.

```
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
```

2. Map the QKD profile to the MACsec policy.

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

3. Apply MACsec policy to the interfaces.

```
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
```

Router 2:

1. Configure the QKD profile.

```
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
```

2. Map the QKD profile to the MACsec policy.

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

3. Apply MACsec policy to the interfaces.

```
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
```

Running Configuration

Router 1:

```
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
!
```

Router 2:

```
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
!t
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
!
```

Verification

```
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, Bob

Peerlist
QKD ID               :Alice
State                :Connected

QKD ID               :Bob
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)
```

Global MACsec Shutdown

The MACsec shutdown feature allows administrator to disable MACsec and re-enable it without modifying the existing MACsec configuration.

Enabling the **macsec shutdown** command, brings down all MACsec sessions on the MACsec-enabled interfaces and resets ports to non-macsec mode. The already existing MACsec configurations remain unaffected by enabling this feature.

Disabling the **macsec shutdown** command, brings up macsec sessions for the configured interfaces and enforces MACsec policy on the port. This feature is disabled by default.

Configure MACsec Shutdown

The following configuration enables the MACsec shutdown on a chassis:

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# macsec shutdown
```

Warning Configuring **macsec shutdown** command disables MACsec on all data ports, system wide. Execute **clear** command to erase cached configuration or **commit** command to continue.

Verify MACsec Shutdown

The **show macsec mka session** command displays a shutdown banner indicating that the MACsec shutdown is enabled.

```
RP/0/RP0/CPU0:router# show macsec mka session
Fri Apr 13 11:56:57.409 IST

********** MACsec shutdown enabled **********
```

The **show macsec mka interface detail** command displays a shutdown banner and the interface-related information.

```
RP/0/RP0/CPU0:fretta-2#show macsec mka interface detail
Fri Apr 13 11:57:02.685 IST
********** MACsec shutdown enabled **********

Number of interfaces on node node0_3_CPU0 : 1
--------------------------------------------------

    Interface Name           : HundredGigE0/3/0/8
    Interface Namestring     : HundredGigE0/3/0/8
    Interface short name     : Hu0/3/0/8
    Interface handle         : 0x1800170
    Interface number         : 0x1800170
    Interface MAC            : 008a.9622.a9d0
    Ethertype                : 888E
    MACsec Shutdown          : TRUE
    Config Received          : TRUE
    IM notify Complete       : TRUE
    Interface CAPS Add       : FALSE
    RxSA CAPS Add            : FALSE
    TxSA CAPS Add            : FALSE
    MKA PSK Info
      Key Chain Name         : kc1
      MKA Cipher Suite       : AES-256-CMAC
      CKN                    : 12 34 56
    MKA fallback_PSK Info
```

```
    fallback keychain Name : fb1
    MKA Cipher Suite       : AES-256-CMAC
    CKN                    : ff ff ff
  Policy                   : *DEFAULT POLICY*
```

Syslog Messages for MACsec Shutdown

The following syslog messages are generated when MACsec shutdown is enabled.

```
%L2-MKA-5-MACSEC_SHUTDOWN_ENABLED : Shutdown ON, disable MACsec on all MACsec enabled ports
%L2-MKA-5-SESSION_STOP            : (Hu0/3/0/8) MKA session stopped,
CKN                               : 123456
%L2-MKA-4-SESSION_UNSECURED       : (Hu0/3/0/8) MKA Session was stopped and is not secured,

CKN                               :123456
%L2-MKA-5-MACSEC_DISABLED         : (Hu0/3/0/8), MACsec disabled (shutdown ON)
```

The following syslog messages are generated when MACsec shutdown is disabled.

```
%L2-MKA-5-MACSEC_SHUTDOWN_DISABLED : Shutdown OFF, resume MACsec on all MACsec enabled ports
%L2-MKA-5-MACSEC_ENABLED                    : (Hu0/3/0/8), MACsec enabled with MUST_SECURE
%L2-MKA-5-SESSION_START                      : (Hu0/3/0/8) MKA session started
CKN                                          :  123456
%L2-MKA-6-MKPDU_ICV_SUCCESS                  : (Hu0/3/0/8), ICV verification success for
RxSCI(008a.9600.60b0/0001), CKN(123456)
%L2-MKA-6-FALLBACK_PSK_MKPDU_ICV_SUCCESS     : (Hu0/3/0/8), ICV verification success for
RxSCI(008a.9600.60b0/0001), CKN(FFFFFF)
%L2-MKA-5-SESSION_SECURED                    : (Hu0/3/0/8) MKA session secured
 CKN                                         :  123456
```

MACsec ISSU

The Cisco IOS XR Software supports in-service software upgrade (ISSU) for Media Access Control Security (MACsec) on the 64-bit IOS XR operating system. This feature allows you to upgrade the network systems without interrupting the secure data connectivity provided by the MACsec session. Such upgrades are feasible if the system and each of its peers support in-service software upgrade.

Commands introduced are:

- suspendFor
- suspendOnRequest

The MACsec ISSU feature is implemented as per the IEEE compliance standard, IEEE Std 802.1Xbx™-2014. It works by suspending the MACsec Key Agreement (MKA) protocol operation temporarily during the ISSU. Once the control plane operation is suspended, the data plane continues to do the encryption with the MACsec hardware keys that are already programmed.

Supported Hardware for MACsec ISSU

The MACsec ISSU feature is supported on Cisco ASR 9000 High Density 100GE Ethernet line cards. The supported hardware variants are:

- A9K-4X100GE-SE
- A9K-8X100GE-SE

- A9K-MPA-1X100GE
- A9K-MPA-2X100GE
- A9K-MPA-20X10GE
- A9K-400G-DWDM-TR

Restrictions for MACsec ISSU

These restrictions apply to MACsec ISSU feature:

- Supported only on 64-bit IOS XR operating system, and on specific hardware (listed in previous section)
- Supported only on pre-shared keys (PSK) based MACsec; not on Extensible Authentication Protocol (EAP) based MACsec. The system terminates the ISSU process if any of the interfaces has EAP MACSec configuration.
- The MACsec ISSU is not supported from release version lower than Cisco IOS XR Software Release 7.1.1 to versions higher or equal toRelease 7.1.1.



Note Disable the MACSec on interfaces or configure **macsec shutdown** command at global configuration mode (if applicable) to run a successful ISSU on the software with release versions lower than Release 7.1.1.

- ISSU is supported only for MACSec sessions running on extended packet numbering (xpn) cipher suites (GCM-AES-XPN-128 and GCM-AES-XPN-256). The system terminates ISSU if there are sessions with non-xpn cipher suites (GCM-AES-128 or GCM-AES-256). The key server selects the cipher suite; the configuration of non-key server cipher suite is insignificant.
- The system terminates MACsec ISSU if there are sessions which are not yet in **suspended** state (use the **show macsec mka session** command to view the session state) after 30 seconds of the load execution phase of ISSU.

Options to Control MKA Protocol Suspension Initiation for ISSU

You can use these two commands under the macsec policy configuration mode to control MKA protocol suspension initiation:

- **suspendFor**: Initiates suspension if it is the key server or requests suspension if it is the non-key server. This option helps admins to control the network by preventing software upgrades that the system triggers without the permission of the key server.
- **suspendOnRequest**: Initiates suspension if it is the key server and when another participant has requested for suspension.

By default, the system enables both options.

Command Usage	Action on the Key Server	Action on the Non-Key Server
suspendFor disable	Disables MKA suspension initiation	Disables the request for MKA suspension
suspendOnRequest disable	Rejects the MKA suspension request from the non-key server	Not applicable

Configuration Example

```
Router#configure
Router(config)#macsec-policy test-policy-mp
/* Disables MKA suspension initiation (if it is the key server) or
disables the request for MKA suspension (if it is the non-key server) */
Router(config-macsec-policy)#suspendFor disable

/* Disables any MKA suspension request from the non-key server */
Router(config-macsec-policy)#suspendOnRequest disable
```

Running Configuration

```
!
macsec-policy test-policy-mp
 suspendFor disable
 suspendOnRequest disable
!
end
```

Verification

A new session state, **SUSPENDED**, is introduced to display the status of MKA suspension operation during ISSU.

```
Router#show macsec mka session
Mon Apr  1 13:13:43.334 IST

NODE: node0_1_CPU0
==============================================================================================

   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN

==============================================================================================

     Hu0/1/0/0       0201.9ab0.85af/0001     1     Suspended     YES      PRIMARY     1234
```

You can use this command to see the details of the MACsec policy:

```
Router#show macsec policy detail
Tue May 21 14:19:31.101 IST
Total Number of Policies = 2
-------------------------------------------------------
Policy Name               : *DEFAULT POLICY*
      Cipher Suite            : GCM-AES-XPN-256
```

```
      Key-Server Priority   : 16
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      Sak Rekey Interval    : OFF
      Include ICV Indicator :  FALSE
      Use Eapol PAE in ICV  : FALSE
      Suspend On Request    : Enabled
      Suspend For           : Enabled

Policy Name                 : test-policy-mp
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 16
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      Sak Rekey Interval    : OFF
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
     Suspend On Request     : Disabled
     Suspend For            : Disabled
```

You can use the **Suspended Peer List** field in the **show macsec mka session detail** command to view the list of peers of the key server that had requested for suspension.

```
Router#show macsec mka session detail
Mon Apr  1 13:13:45.893 IST
NODE: node0_1_CPU0
 MKA Detailed Status for MKA Session
===================================
Status: SUSPENDED - Secured MACsec with suspended MKA operations

Local Tx-SCI                    : 0201.9ab0.85af/0001
Local Tx-SSCI                   : 2
Interface MAC Address           : 0201.9ab0.85af
MKA Port Identifier             : 1
Interface Name                  : Hu0/1/0/0
CAK Name (CKN)                  : 1234
CA Authentication Mode          : PRIMARY-PSK
Keychain                        : kc1
Member Identifier (MI)          : 89E20E40ACED97317596CCC0
Message Number (MN)             : 156
Authenticator                   : NO
Key Server                      : YES
MKA Cipher Suite                : AES-256-CMAC
Configured MACSec Cipher Suite  : GCM-AES-XPN-256

Latest SAK Status               : Rx & Tx
Latest SAK AN                   : 2
Latest SAK KI (KN)              : 89E20E40ACED97317596CCC000000001 (1)
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         : 120s

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          SSCI  KS-Priority
-----------------------------------------------------------------------------
DA057FA6983845205FD0EB28      162       0257.3fae.5cda/0001    1        16

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

Suspended Peer List:
-----------------------------------------------------------------------------
       Rx-SCI          SSCI
-----------------------------------------------------------------------------
  02573fae5cda0001      1
Peers Status:
Last Tx MKPDU           : 2019 Apr 01 13:13:45.350
Peer Count              : 1

RxSCI                   : 02573FAE5CDA0001
  MI                    : DA057FA6983845205FD0EB28
  Peer CAK              : Match
  Latest Rx MKPDU       : 2019 Apr 01 13:13:44.238
```

Also, these SYSLOGS indicate various stages of the ISSU process on the key server and the non-key server:

- **L2-MKA-5-SUSPENSION-REQUESTED**
  - On the non-key server—when it requests for suspension. (ISSU)

```
(Hu0/1/0/0), Requesting suspension of MACsec control plane operation
```

- **L2-MKA-5-SUSPENSION-START-REQUEST_RECEIVED**
  - On the key server—when it receives non-zero value for the suspendFor parameter from the non-key server. The key server accepts or rejects the suspension request based on the value configured for the **suspendOnRequest** command.

```
(Hu0/1/0/0), MACsec control plane operation suspension start request from
```

```
Peer(02573fae5cda0001) accepted.
```

or

```
(Hu0/1/0/0), MACsec control plane operation suspension start request from
Peer(02573fae5cda0001) rejected (policy conflict).
```

- **L2-MKA-5-SUSPENSION-START**
  - On the key server—when it initiates suspension.
  - On the non-key server—when it receives non-zero value for the **suspendFor** parameter from the key server.

```
(Hu0/1/0/0), MACsec control plane operation suspended.
```

- **L2-MKA-5-SUSPENSION-STOP-REQUEST_RECEIVED**
  - On the key server—when it receives a zero value for the**suspendFor** parameter from the peer which had previously requested for suspension.

```
(Hu0/1/0/0), MACsec control plane operation suspension stop received from
Peer(02573fae5cda0001)
```

- **L2-MKA-5-SUSPENSION-STOP**
  - On the key server—when it terminates the suspension.
  - On the non-key server—when it receives a zero value for the **suspendFor** parameter from the key server.

```
(Hu0/1/0/0), MACsec control plane operation resumed
```

Related Topics

MACsec ISSU, on page 94

Associated Commands

- suspendFor
- suspendOnRequest

MACsec SNMP MIB (IEEE8021-SECY-MIB)

Table 12: Feature History Table

Feature Name	Release Information	Description
MACsec SNMP MIB (IEEE8021-SECY-MIB) support on A99-10X400GE-X-SE	Release 7.5.3	With this feature, we have enabled IEEE8021-SECY-MIB capabilities on the A99-10X400GE-X-SE line card. The IEEE8021-SECY-MIB enables the user to query on the SecY (Security Entity) data, encryption, decryption, and hardware statistics.

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 is an entity that operates the MAC Security protocol in the router. 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 13: 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

References:

- For more technical details on MACsec SNMP MIB (IEEE8021-SECY-MIB) support in ASR 9000 Series Routers, download the *IEEE8021-SECY-MIB* from MIB Locator in Cisco Feature Navigator.
- For more information on the Simple Network Management (SNMP) Protocol, see chapter *Configuring Simple Network Management Protocol* in *System Management Configuration Guide for Cisco 8000 Series Routers*.
- For more information on the IEEE8021-SecY-MIB, see the following URL:http://www.ieee802.org/1/files/public/MIBs/IEEE8021-SECY-MIB-200601100000Z.mib