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.

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.

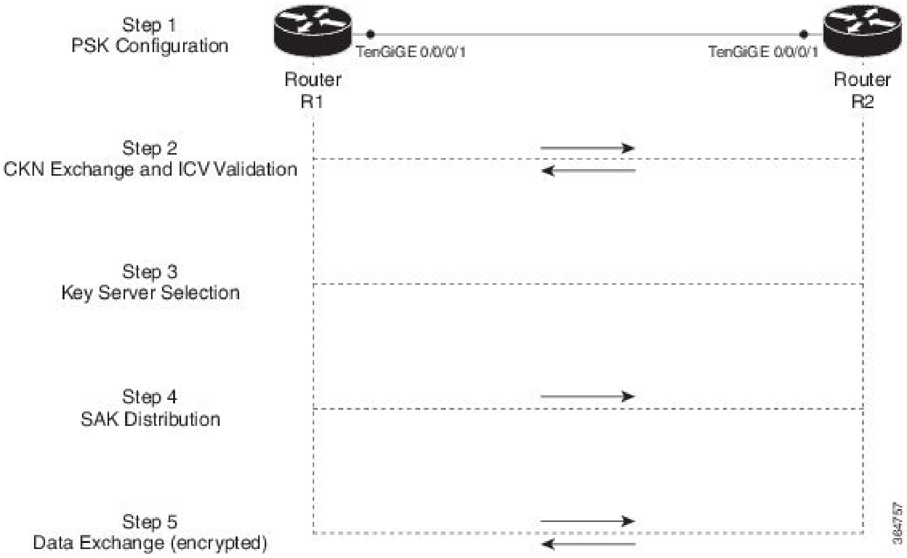
MACservice can be deployed in the network as a technology or as a service. For more information, see Types of MACsec Implementation, on page 3

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.

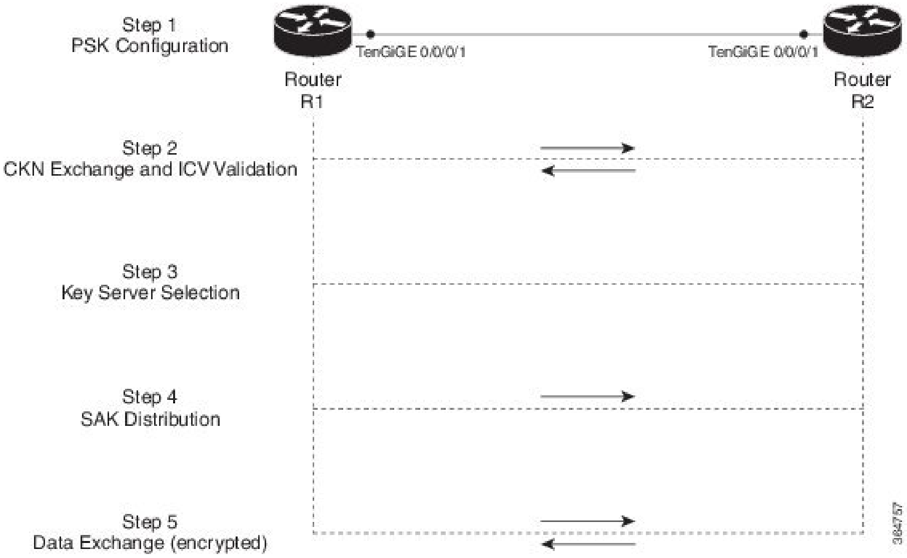
  For more information on configuring MACsec, see *Creating a MACsec Keychain* and *Creating a MACsec Policy*

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.

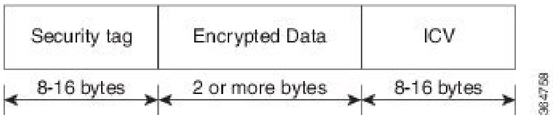
MACsec Frame Format

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



Configuring and Verifying MACSec Encryption

MACSec can be configured on physical ethernet interfaces, VLAN sub-interfaces, or interface bundles (link bundles), as explained in this section.

Note MACSec on a VLAN sub-interface is configured in same way as on a physical interface. For a successful MKA session to be up on any VLAN sub-interface, it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet sub-interfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined. The sub-interfaces belonging to a physical interface can have 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

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.

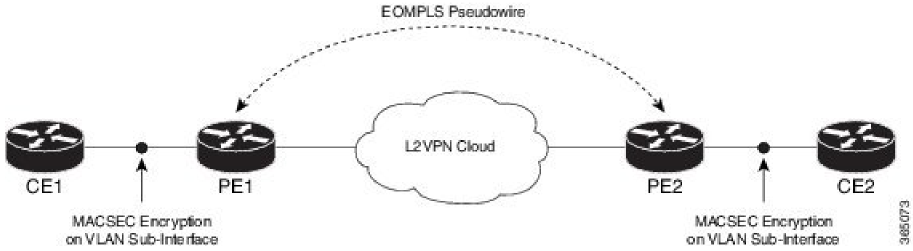
The following table lists the number of sub-interfaces with MACSec supported in a L2VPN.

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

Table 1: Supported MACSec Sessions on Sub-Interfaces

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

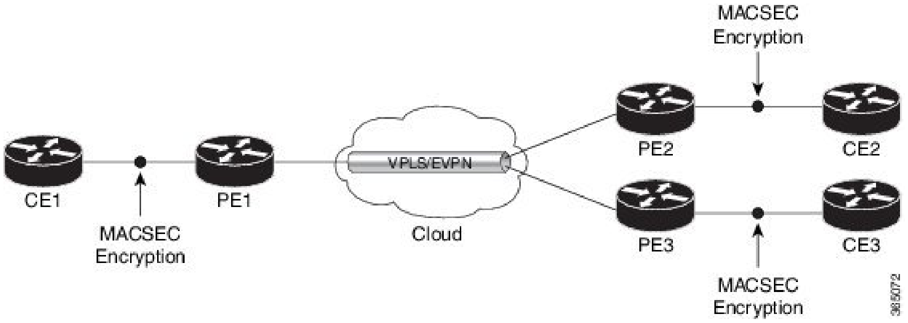
Figure 4: MACSec in a L2VPN Cloud


EOMPLS Pseudowire
L2VPN Cloud
CE1
PE1
PE2
CE2
MACSEC Encryption on VLAN Sub-Interface
MACSEC Encryption on VLAN Sub-Interface
365073

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



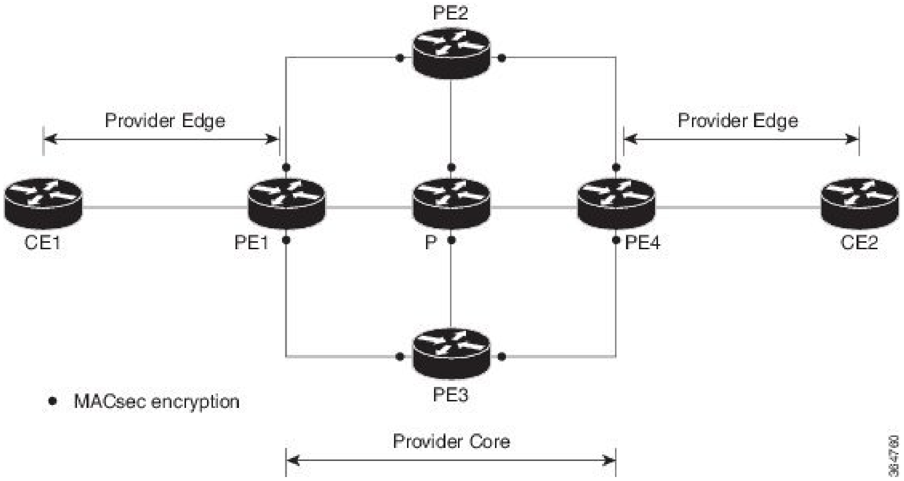
Use Case 3: MACSec in an MPLS Core Network

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.

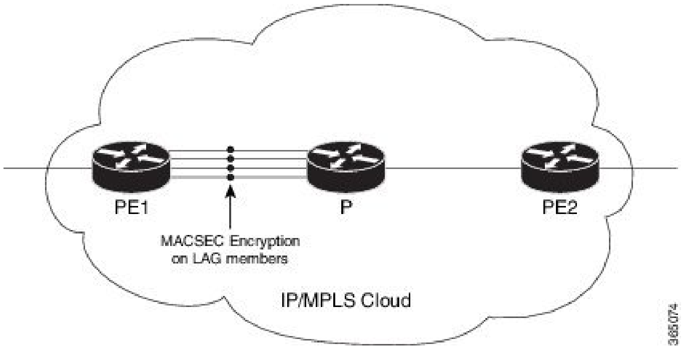
Figure 6: MACSec on Physical Interfaces in an MPLS Core Network



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



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

Prior to configuring MACSec on a router interface, the MACSec key chain and MACSec policy must be defined. Configuring MACSec encryption involves the following steps:

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

Creating a MACsec Keychain

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

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

Guidelines for Configuring MACsec Keychain

MACsec keychain management has the following configuration guidelines:

- 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.

SUMMARY STEPS

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

DETAILED STEPS

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)# keychain mac_chain
```

Step 2 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.

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

Step 3 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 1234567812345678123456781234567812345678123456781234567812345678 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 4 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 5 Commit your configuration.

Example:

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

This completes the configuration of the MACsec keychain.

Prerequisites for Configuring MACSec on Bundle Member Interfaces

To enable MACSec on bundle members, an user-defined policy must be configured with Should-Secure policy, or Must-Secure policy with **policy-exception LACP-in-clear** command.

Note By default, the system uses the Must-Secure security policy.

Example: Configuring MACSec on Bundle Member With Should-Secure Policy

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


sh runn macsec-policy should-secure
  macsec-policy should-secure
  security-policy should-secure
!

router(config)# interface HundredGigE 0/1/1/1 # Applying the Should-Secure MACSec Policy
```

```
on Bundle Member Interface
  router(config-if)# bundle id 12 mode active
  router(config-if)# macsec psk-keychain kc1 policy should-secure
```

Example: Configuring MACSec on Bundle Member With Must-Secure Policy

```
(config)#macsec-policy must-secure
(config-macsec-policy)#security-policy must-secure
(config-macsec-policy)#policy-exception lacp-in-clear
(config-macsec-policy)#commit

#sh runn macsec-policy must-secure
macsec-policy must-secure
  security-policy must-secure
  policy-exception lacp-in-clear
!
router(config)# interface HundredGigE 0/1/1/2 #Applying the Must-Secure MACSec Policy on
Bundle Member Interface
  router(config-if)# bundle id 12 mode active
  router(config-if)# macsec psk-keychain kc1 policy must-secure
```

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

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
```

Note In this example, we have used the GCM-AES-XPN-256 encryption algorithm. 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.

Step 3 Configure the confidentiality offset for MACsec encryption.

Example:

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

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 2: 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.

Applying MACsec Configuration on an Interface

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

Guidelines for MACsec Interface Configuration

Following are the guidelines for configuring MACsec interface:

- 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.

SUMMARY STEPS

1. Enter the global configuration mode.
2. Enter the interface configuration mode.
3. Apply the MACsec configuration on an interface.
4. Commit your configuration.

DETAILED STEPS

Step 1 Enter the global configuration mode.

Example:

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

Step 2 Enter the interface configuration mode.

Example:

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

Step 3 Apply the MACsec configuration on an interface.

MACsec PSK Configuration

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

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec psk-keychain mac_chain policy mac_policy
RP/0/RSP0/CPU0:router(config-if)# exit
```

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

Example:

```
RP/0/RSP0/CPU0:router(config-if)# macsec psk-keychain script_key_chain2
RP/0/RSP0/CPU0:router(config-if)# exit
```

Step 4 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.

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 Verify the MACsec policy configuration.

Example:

```
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:

```
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         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
```

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 **#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.

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

Example:

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

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

Step 5 Verify the MACsec session counter statistics.

Example:

```
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

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 : 0xbfffffb8
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.