MACsec Encryption

This chapter contains the following sections:

MACsec and the MACsec Key Agreement (MKA) Protocol

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The switch supports 802.1AE encryption with MACsec Key Agreement (MKA) on on switch-to-host links for encryption between the switch and host device. The switch also supports MACsec encryption for switch-to-switch (inter-network device) security using MKA-based key exchange protocol. The MKA protocol provides the required session keys and manages the required encryption keys.

Note When switch-to-switch MACsec is enabled, all traffic is encrypted except EAP-over-LAN (EAPOL) packets.

Important On the ESS-3300, MACsec is supported on 1 gigabit ethernet downlink ports only.

Link layer security can include both packet authentication between switches and MACsec encryption between switches (encryption is optional).

Table 1: MACsec Support on Switch Ports

Connections	MACsec support
Switch-to-host	MACsec MKA encryption
Switch-to-switch	MACsec MKA encryption

Cisco TrustSec is meant only for switch-to-switch links and is not supported on switch ports connected to end hosts, such as PCs or IP phones. MKA is supported on switch-to-host facing links as well as switch-to-switch links. Host-facing links typically use flexible authentication ordering for handling heterogeneous devices with or without IEEE 802.1x, and can optionally use MKA-based MACsec encryption.

Network Edge Access Topology (NEAT) is used for compact switches to extend security outside the wiring closet.

MACsec and MACsec Key Agreement (MKA) are implemented after successful authentication using certificate-based MACsec or Pre Shared Key (PSK) framework.

MKA Policies

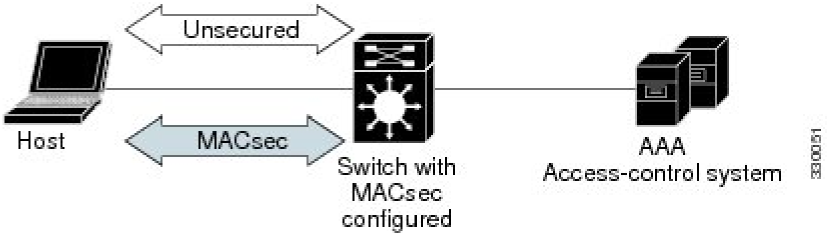
To enable MKA on an interface, a defined MKA policy should be applied to the interface. You can configure these options:

- Policy name, not to exceed 16 ASCII characters.
- Confidentiality (encryption) offset of 0, 30, or 50 bytes for each physical interface

Single-Host Mode

The figure shows how a single EAP authenticated session is secured by MACsec by using MKA.

Figure 1: MACsec in Single-Host Mode with a Secured Data Session



Switch-to-Switch MKA MACsec Must Secure Policy

When MACsec is enabled on an interface, all interface traffic except EAPoL traffic is secured by default ("must-secure" is the default) on both the ingress and the egress. Unencrypted packets are dropped until the MKA session is secured. However, to enable MACsec on selected interfaces, you can choose to allow unencrypted packets to be transmitted or received from the same physical interface by setting **macsec access-control** to **should-secure**. This option allows unencrypted traffic to flow until the MKA session is secured. After the MKA session is secured, only encrypted traffic can flow. For configuration details, see Configuring MACsec MKA on an Interface using PSK, on page 16.

MKA/MACsec for Port Channel

MKA/MACsec can be configured on the port members of a port channel. MKA/MACsec is agnostic to the port channel since the MKA session is established between the port members of a port channel.

Note Etherchannel links that are formed as part of the port channel can either be congruent or disparate i.e. the links can either be MACsec-secured or non-MACsec-secured. MKA session between the port members is established even if a port member on one side of the port channel is not configured with MACsec.

We recommend that you enable MKA/MACsec on all the member ports for better security of the port channel.

MACsec Cipher Announcement

Cipher Announcement allows the supplicant and the authenticator to announce their respective MACsec Cipher Suite capabilities to each other. Both the supplicant and the authenticator calculate the largest common supported MACsec Cipher Suite and use the same as the keying material for the MKA session.

Note Only the MACsec Cipher Suite capabilities which are configured in the MKA policy are announced from the authenticator to the supplicant.

There are two types of EAPoL Announcements:

- Unsecured Announcements (EAPoL PDUs) : Unsecured announcements are EAPoL announcements carrying MACsec Cipher Suite capabilities in an unsecured manner. These announcements are used to decide the width of the key used for MKA session prior to authentication.
- Secure Announcements (MKPDUs) : Secure announcements revalidate the MACsec Cipher Suite capabilities which were shared previously through unsecure announcements.

Once the session is authenticated, peer capabilities which were received through EAPoL announcements are revalidated with the secure announcements. If there is a mismatch in the capabilities, the MKA session tears down.

Limitations for MACsec Cipher Announcement

- MACsec Cipher Announcement is supported only on the switch-to-host links.
- The MKA session between the supplicant and the authenticator does not tear down even if the MACsec Cipher Suite capabilities configured on both do not result in a common cipher suite.

MKA Statistics

Some MKA counters are aggregated globally, while others are updated both globally and per session. You can also obtain information about the status of MKA sessions.

This is an example of the show mka statistics command output:

```
Switch# show mka sessions

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

==========================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
==========================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO                YES
43             c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000

Switch#show mka sessions interface G1/0/1
```

```
Summary of All Currently Active MKA Sessions on Interface GigabitEthernet1/0/1...

==========================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
==========================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO                YES
43             c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000


Switch#show mka sessions interface G1/0/1 de

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

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)........... 0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89567
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... D46CBEC05D5D67594543CEAE00000001 (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.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
Send Secure Announcement.. DISABLED
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 Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
  38046BA37D7DA77E06D006A9  89555       c800.8459.e764/002a   10

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

```
Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------

Switch#show mka sessions de
Switch#show mka sessions detail

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

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)........... 0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... D46CBEC05D5D67594543CEAE00000001 (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.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
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 Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
  38046BA37D7DA77E06D006A9  89560       c800.8459.e764/002a   10

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

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------

Switch#sh mka pol

MKA Policy Summary...
```

```
Policy            KS       Delay   Replay  Window Conf   Cipher          Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)        Applied
==========================================================================================
*DEFAULT POLICY*  0        FALSE   TRUE    0      0      GCM-AES-128

p1                1        FALSE   TRUE    0      0      GCM-AES-128

p2                2        FALSE   TRUE    0      0      GCM-AES-128     Gi1/0/1


Switch#sh mka poli
Switch#sh mka policy p2
Switch#sh mka policy p2 ?
  detail    Detailed configuration/information for MKA Policy
  sessions  Summary of all active MKA Sessions with policy applied
  |         Output modifiers
  <cr>

Switch#sh mka policy p2 de

MKA Policy Configuration ("p2")
========================
MKA Policy Name........ p2
Key Server Priority.... 2
Confidentiality Offset. 0
Send Secure Announcement..DISABLED
Cipher Suite(s)........ GCM-AES-128

Applied Interfaces...
  GigabitEthernet1/0/1

Switch#sh mka policy p2

MKA Policy Summary...

Policy            KS       Delay   Replay  Window Conf   Cipher          Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)        Applied
==========================================================================================
p2                2        FALSE   TRUE    0      0      GCM-AES-128     Gi1/0/1


Switch#sh mka se?
sessions

Switch#sh mka ?
  default-policy  MKA Default Policy details
  keychains       MKA Pre-Shared-Key Key-Chains
  policy          MKA Policy configuration information
  presharedkeys   MKA Preshared Keys
  sessions        MKA Sessions summary
  statistics      Global MKA statistics
  summary         MKA Sessions summary & global statistics

Switch#sh mka statis
Switch#sh mka statistics ?
  interface  Statistics for a MKA Session on an interface
  local-sci  Statistics for a MKA Session identified by its Local Tx-SCI
  |          Output modifiers
  <cr>

Switch#sh mka statistics inter
Switch#show mka statistics interface G1/0/1

MKA Statistics for Session
```

```
==========================
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.......... 1
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 1

MKPDU Statistics
   MKPDUs Validated & Rx... 89585
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Transmitted...... 89596
      "Distributed SAK".. 1
      "Distributed CAK".. 0


Switch#show mka ?
  default-policy  MKA Default Policy details
  keychains       MKA Pre-Shared-Key Key-Chains
  policy          MKA Policy configuration information
  presharedkeys   MKA Preshared Keys
  sessions        MKA Sessions summary
  statistics      Global MKA statistics
  summary         MKA Sessions summary & global statistics

Switch#show mka summ
Switch#show mka summary

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

==========================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
==========================================================================================
Gi1/0/1       204c.9e85.ede4/002b p2               NO                YES
43            c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000



MKA Global Statistics
=====================
MKA Session Totals
   Secured.................... 1
   Reauthentication Attempts.. 0

   Deleted (Secured).......... 0
   Keepalive Timeouts......... 0

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

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

MKPDU Statistics
   MKPDUs Validated & Rx...... 89589
      "Distributed SAK"..... 0
      "Distributed CAK"..... 0
   MKPDUs Transmitted......... 89600
      "Distributed SAK"..... 1
      "Distributed CAK"..... 0

MKA Error Counter Totals
========================
Session Failures
   Bring-up Failures................ 0
   Reauthentication Failures........ 0
   Duplicate Auth-Mgr Handle........ 0

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

CA Failures
   Group CAK Generation............. 0
   Group CAK Encryption/Wrap........ 0
   Group CAK Decryption/Unwrap...... 0
   Pairwise CAK Derivation.......... 0
   CKN Derivation................... 0
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0
MACsec Failures
   Rx SC Creation................... 0
   Tx SC Creation................... 0
   Rx SA Installation............... 0
   Tx SA Installation............... 0

MKPDU Failures
   MKPDU Tx......................... 0
   MKPDU Rx Validation.............. 0
   MKPDU Rx Bad Peer MN............. 0
   MKPDU Rx Non-recent Peerlist MN.. 0

Switch#
```

Certificate Based MACsec

The Certificate based MACsec Encryption feature uses 802.1X port-based authentication with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) to carry Certificates for ports where MACsec encryption is required. EAP-TLS mechanism is used for the mutual authentication and to get the Master Session Key (MSK) from which the Connectivity Association Key (CAK) is derived for the MACsec Key Agreement (MKA) protocol.

This feature allows keys to be managed at a centralized server (CA) over PSK (Pre-Shared Key) based MACsec. Switch to switch MACsec is supported. See Configuring Certificate Based MACsec, on page 18 for more information.

How to Configure MACsec Encryption

Limitations and Restrictions

MACsec has these limitations and restrictions:

- Ports should be in access mode or trunk mode.
- MKA is not supported on port-channels. Individual links that comprise the port-channel can use MACsec.
- High Availability for MKA is not supported.
- Ports with **no switchport** are not supported.
- ESS3300 uplink ports do not have a PHY and hence do not support MACSec.
- Certificate-based MACsec is supported only if the access-session is configured as closed or in multiple-host mode. None of the other configuration modes are supported.

Prerequisites for MACsec Encryption

Prerequisites for MACsec Encryption:

- Ensure that 802.1x authentication and AAA are configured on your device.

Configuring MKA and MACsec

Default MACsec MKA Configuration

MACsec is disabled. No MKA policies are configured.

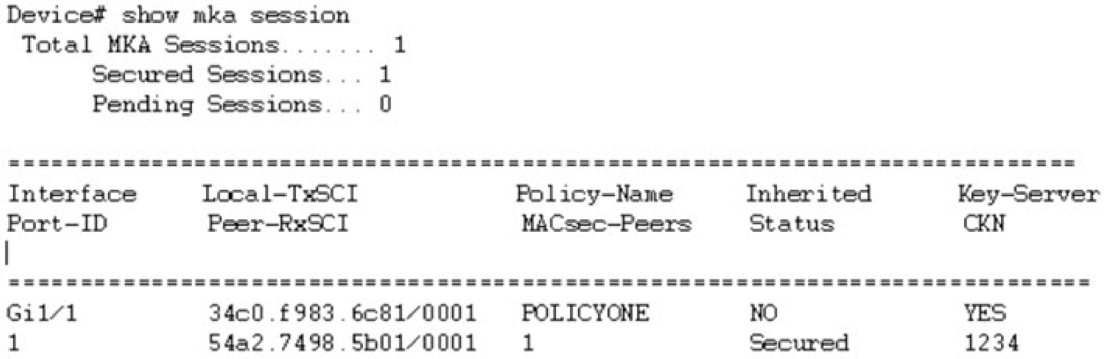
MKA-PSK: CKN Behavior Change

A change was made in Cisco IOS XE from how the CKN (the "key") was implemented in Cisco IOS Classic. When an IE switch running Cisco IOS XE needs to make a PreShared Key (PSK) MACSec connection with an IE switch running Cisco IOS Classic, the configured "key" value must be 64 hex characters long. Also, the "key" value must match the same on the IE switch running Cisco IOS Classic. The same "key" value on the Cisco IOS Classic side does not have to pad zeros.

This Cisco IOS XE example shows key chain configuration when connecting two Cisco IOS XE devices:

```
configure terminal
key chain KEYCHAINONE macsec
key 1234
cryptographic-algorithm aes-128-cmac
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
end
```

For the above example, following is the output for the two Cisco IOS XE connected devices for the **show mka session** command:



```
Device# show mka session
 Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

===========================================================================
Interface     Local-TxSCI           Policy-Name     Inherited      Key-Server
Port-ID       Peer-RxSCI            MACsec-Peers    Status         CKN
|
============================================================================
Gi1/1         34c0.f983.6c81/0001   POLICYONE       NO             YES
1             54a2.7498.5b01/0001   1               Secured        1234
```

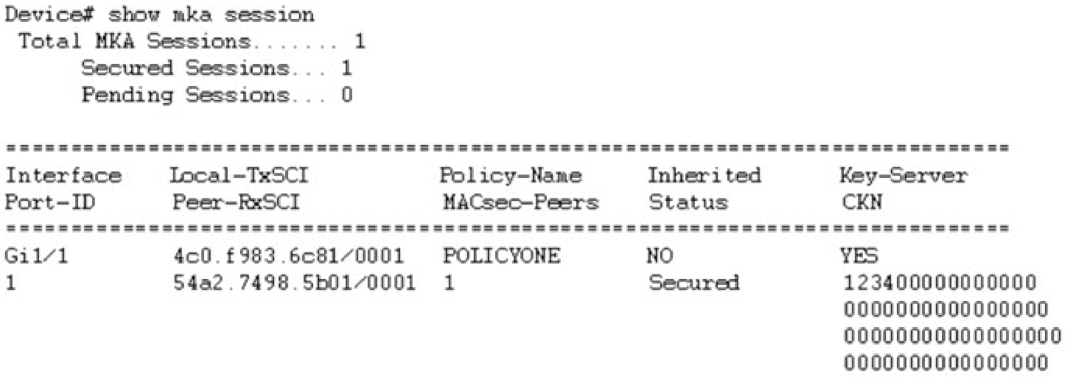
Note that the CKN key-string is exactly the same that has been configured for the key as hex-string. This is an example of Cisco IOS XE to Cisco IOS XE PSK where the CKN is not zero padded.

For interoperability between devices running Cisco IOS XE and devices running Cisco IOS Classic, the Cisco IOS XE devices must zero pad the CKN value (the "key") to match the Cisco IOS Classic where the configuration CKN key does not have to be zero padded. Cisco IOS Classic zero pads the CKN key value for the user, adding zeroes after the configured CKN value to make the length 64 hex characters long. The CKN key value must be 64 hex characters long.

The following example shows configuration of the CKN key-string on a Cisco IOS Classic device:

```
config t
key chain KEYCHAINONE macsec
key 1234
cryptographic-algorithm aes-128-cmac
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
```

For the above example, following is the output on the Cisco IOS Classic device for the **show mka session** command:



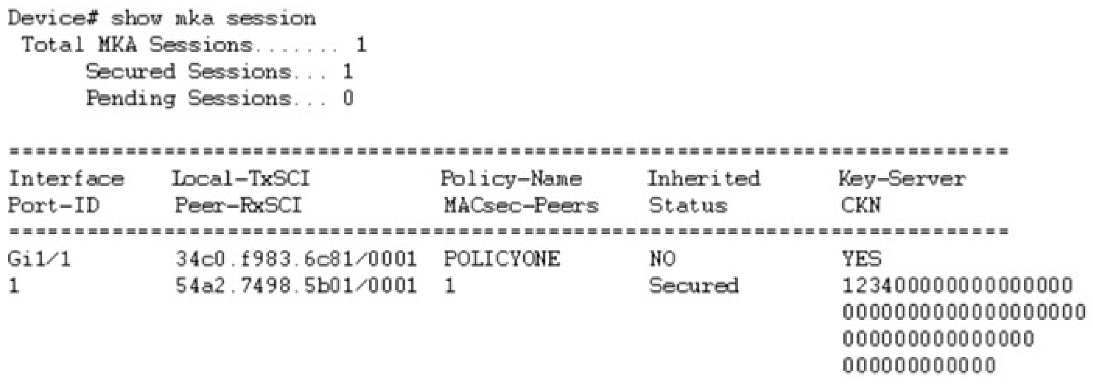
```
Device# show mka session
 Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

========================================================================
Interface    Local-TxSCI          Policy-Name     Inherited     Key-Server
Port-ID      Peer-RxSCI           MACsec-Peers    Status        CKN
========================================================================
Gi1/1        4c0.f983.6c81/0001   POLICYONE       NO            YES
1            54a2.7498.5b01/0001  1               Secured       123400000000000
                                                                0000000000000000
                                                                00000000000000000
                                                                0000000000000000
```

This example shows the configuration on the Cisco IOS XE device for interoperability with Cisco IOS Classic devices:

```
config t
key chain KEYCHAINONE macsec
key 1234000000000000000000000000000000000000000000000000000000000000
cryptographic-algorithm aes-128-cmac
```

```
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
```

For the above example, following is the **show mka session** output on the Cisco IOS XE device:



```
Device# show mka session
 Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

==============================================================================
Interface    Local-TxSCI          Policy-Name     Inherited      Key-Server
Port-ID      Peer-RxSCI           MACsec-Peers    Status         CKN
==============================================================================
Gi1/1        34c0.f983.6c81/0001  POLICYONE       NO             YES
1            54a2.7498.5b01/0001  1               Secured        123400000000000000
                                                                 0000000000000000000
                                                                 000000000000000
                                                                 000000000000
```

Configuring an MKA Policy

SUMMARY STEPS

1. **configure terminal**
2. **mka policy** *policy name*
3. **send-secure-announcements**
4. **key-server** *priority*
5. **include-icv-indicator**
6. **macsec-cipher-suite** *gcm-aes-128*
7. **confidentiality-offset** *Offset value*
8. **end**
9. **show mka policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mka policy** *policy name*	Identify an MKA policy, and enter MKA policy configuration mode. The maximum policy name length is 16 characters. **Note** The default MACsec cipher suite in the MKA policy will always be "GCM-AES-128". If the device supports both "GCM-AES-128" and "GCM-AES-256" ciphers, it is highly recommended to define and use a user defined MKA policy to include both 128 and 256 bits ciphers or only 256 bits cipher, as may be required.
Step 3	**send-secure-announcements**	Enabled secure announcements.

	Command or Action	Purpose
		Note By default, secure announcements are disabled.
Step 4	**key-server** *priority*	Configure MKA key server options and set priority (between 0-255). **Note** When value of key server priority is set to 255, the peer can not become the key server. The key server priority value is valid only for MKA PSK; and not for MKA EAPTLS.
Step 5	**include-icv-indicator**	Enables the ICV indicator in MKPDU. Use the no form of this command to disable the ICV indicator — **no include-icv-indicator**.
Step 6	**macsec-cipher-suite** *gcm-aes-128*	Configures cipher suite for deriving SAK with 128-bit encryption.
Step 7	**confidentiality-offset** *Offset value*	Set the Confidentiality (encryption) offset for each physical interface **Note** Offset Value can be 0, 30 or 50. If you are using Anyconnect on the client, it is recommended to use Offset 0.
Step 8	**end**	Returns to privileged EXEC mode.
Step 9	**show mka policy**	Verify your entries.

Example

This example configures the MKA policy:

```
Switch(config)# mka policy mka_policy
Switch(config-mka-policy)# key-server priority 200
Switch(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Switch(config-mka-policy)# confidentiality-offset 30
Switch(config-mka-policy)# end
```

Configure Switch-to-host MACsec Encryption

Follow these steps to configure MACsec on an interface with one MACsec session for voice and one for data:

SUMMARY STEPS

1. **enable**
2. **configureterminal**
3. **interface** *type number*
4. **switchport access vlan***vlan-id*
5. **switchport mode access**
6. **macsec**

7. **authentication event linksec fail action authorize vlan** *vlan-id*
8. **authentication host-mode multi-domain**
9. **authentication linksec policy must-secure**
10. **authentication port-control auto**
11. **authentication periodic**
12. **authentication timer reauthenticate**
13. **authentication violation protect**
14. **mka policy** *policy-name*
15. **dot1x pae authenticator**
16. **spanning-tree portfast**
17. **end**
18. **show authentication session interface** *interface-id*
19. **show mka sessions**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter the password if prompted.
Step 2	**configureterminal** **Example:** `Device>configure terminal`	Enters the global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	**switchport access vlan***vlan-id* **Example:** `Device(config-if)# switchport access vlan 1`	Configures the access VLAN for the port.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Configures the interface as an access port.
Step 6	**macsec** **Example:** `Device(config-if)# macsec`	Enables 802.1ae MACsec on the interface. The macsec command enables MKA MACsec on switch-to-host links only.
Step 7	**authentication event linksec fail action authorize vlan** *vlan-id* **Example:** `Device(config-if)# authentication event linksec fail action authorize vlan 1`	(Optional) Specifies that the switch processes authentication link-security failures resulting from unrecognized user credentials by authorizing a restricted VLAN on the port after a failed authentication attempt.

	Command or Action	Purpose
Step 8	**authentication host-mode multi-domain** **Example:** `Device(config-if)# authentication host-mode multi-domain`	Configures authentication manager mode on the port to allow both a host and a voice device to be authenticated on the 802.1x-authorized port. If not configured, the default host mode is single.
Step 9	**authentication linksec policy must-secure** **Example:** `Device(config-if)# authentication linksec policy must-secure`	Sets the LinkSec security policy to secure the session with MACsec if the peer is available. If not set, the default is *should secure*.
Step 10	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	Enables 802.1x authentication on the port. The port changes to the authorized or unauthorized state based on the authentication exchange between the switch and the client.
Step 11	**authentication periodic** **Example:** `Device(config-if)# authentication periodic`	(Optional) Enables or disables re-authentication for this port .
Step 12	**authentication timer reauthenticate** **Example:** `Device(config-if)# authentication timer reauthenticate`	(Optional) Enters a value between 1 and 65535 (in seconds). Obtains re-authentication timeout value from the server. Default re-authentication time is 3600 seconds.
Step 13	**authentication violation protect** **Example:** `Device(config-if)# configure terminal`	Configures the port to drop unexpected incoming MAC addresses when a new device connects to a port or when a device connects to a port after the maximum number of devices are connected to that port. If not configured, the default is to shut down the port.
Step 14	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy mka_policy`	Applies an existing MKA protocol policy to the interface, and enable MKA on the interface. If no MKA policy was configured (by entering the**mka policy** global configuration command).
Step 15	**dot1x pae authenticator** **Example:** `Device(config-if)# dot1x pae authenticator`	Configures the port as an 802.1x port access entity (PAE) authenticator.
Step 16	**spanning-tree portfast** **Example:** `Device(config-if)# spanning-tree portfast`	Enables spanning tree Port Fast on the interface in all its associated VLANs. When the Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes
Step 17	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 18	**show authentication session interface** *interface-id* **Example:** `Device# show authentication session interface GigabitEthernet 1/0/1`	Verifies the authorized session security status.
Step 19	**show mka sessions** **Example:** `Device# show mka sessions`	Verifies the established MKA sessions.

Configuring MACsec MKA using Pre Shared Key (PSK)

SUMMARY STEPS

1. **configure terminal**
2. **key chain** *key-chain-name* **macsec**
3. **key** *hex-string*
4. **cryptographic-algorithm** {*gcm-aes-128* | *gcm-aes-256*}
5. **key-string** { [0|6|7] *pwd-string* | *pwd-string*}
6. **lifetime local** [*start timestamp {hh::mm::ss* | *day* | *month* | *year}*] [**duration** *seconds* | *end timestamp {hh::mm::ss* | *day* | *month* | *year*}]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**key chain** *key-chain-name* **macsec**	Configures a key chain and enters the key chain configuration mode.
Step 3	**key** *hex-string*	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. **Note** For 128-bit encryption, use 32 hex digit key-string. For 256-bit encryption, use 64 hex digit key-string.
Step 4	**cryptographic-algorithm** {*gcm-aes-128*	*gcm-aes-256*}	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 5	**key-string** { [0	6	7] *pwd-string*	*pwd-string*}	Sets the password for a key string. Only hex characters must be entered..
Step 6	**lifetime local** [*start timestamp {hh::mm::ss*	*day*	*month*	*year}*] [**duration** *seconds*	*end timestamp {hh::mm::ss*	*day*	*month*	*year*}]	Sets the lifetime of the pre shared key.

	Command or Action	Purpose
Step 7	**end**	Returns to privileged EXEC mode.

Example

Following is an indicative example:

```
Switch(config)# Key chain keychain1 macsec
Switch(config-key-chain)# key 1000
Switch(config-keychain-key)# cryptographic-algorithm gcm-aes-128
Switch(config-keychain-key)# key-string 12345678901234567890123456789012
Switch(config-keychain-key)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016
Switch(config-keychain-key)# end
```

Configuring MACsec MKA on an Interface using PSK

Note To avoid traffic drop across sessions, the **mka policy** command must be configured before the **mka pre-shared-key key-chain** command.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **macsec access-control should-secure**
4. **macsec**
5. **mka policy** *policy-name*
6. **mka pre-shared-key key-chain** *key-chain name*
7. **macsec replay-protection window-size** *frame number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode.
Step 3	**macsec access-control should-secure**	(Optional) Allows unencrypted traffic to flow until the MKA session is secured. After the MKA session is secured, only encrypted traffic can flow. By default, traffic is dropped until the MKA session is secured. To revert to the default behavior, use the **no macsec access-control should-secure** command.
Step 4	**macsec**	Enables MACsec on the interface.

	Command or Action	Purpose
Step 5	**mka policy** *policy-name*	Configures an MKA policy.
Step 6	**mka pre-shared-key key-chain** *key-chain name*	Configures an MKA pre-shared-key key-chain name.
Step 7	**macsec replay-protection window-size** *frame number*	Sets the MACsec window size for replay protection.
Step 8	**end**	Returns to privileged EXEC mode.

Example

The following example configures an MKA policy and an MKA pre-shared-key key-chain name, and sets the MACsec window size for replay protection:

```
Switch(config)# interface GigabitEthernet 1/1
Switch(config-if)# mka policy mka_policy
Switch(config-if)# mka pre-shared-key key-chain key-chain-name
Switch(config-if)# macsec replay-protection window-size 10
Switch(config-if)# end
```

Note It is not recommended to change the MKA policy on an interface with MKA PSK configured when the session is running. However, if a change is required, you must reconfigure the policy as follows:

1. Disable the existing session by removing macsec configuration on each of the participating nodes using the **no macsec** command.
2. Configure the MKA policy on the interface on each of the participating nodes using the **mka policy policy-name** command.
3. Enable the new session on each of the participating node by using the **macsec** command.

The following examples show how to configure the interface to use **should-secure** instead of the default **must-secure** and how to change it back to the default **must-secure**.

Note Modifying **access-control** is not allowed when the session is up and running. You first need to remove the MACsec configuration by using the **no macsec** command, and then configure **access-control**.

Example 1: To change from **must-secure** to **should-secure**:

```
Switch(config-if)#no macsec
Switch(config-if)#macsec access-control should-secure
Switch(config-if)#macsec // this switches the access-control from must-secure & restarts
the macsec session with new behaviour.
```

Example 2: To change from **should-secure** to **must-secure**:

```
Switch(config-if)#no macsec
Switch(config-if)#no macsec access-control
Switch(config-if)#macsec
```

Configuring Certificate Based MACsec

To configure MACsec with MKA on point-to-point links, perform these tasks:

- Generating Key Pairs
- Configuring Enrollment using SCEP
- Configuring Enrollment Manually
- Enabling 802.1x Authentication and Configuring AAA, on page 23
- Configuring EAP-TLS Profile and 802.1x Credentials, on page 25
- Applying the 802.1x MKA MACsec Configuration on Interfaces, on page 27

Prerequisites for Certificate Based MACsec

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate or obtain a third-party certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE).
- Ensure that 802.1x authentication and AAA are configured on your device.

Generating Key Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa label** *label-name* **general-keys modulus** *size*
4. **end**
5. **show authentication session interface** *interface-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa label** *label-name* **general-keys modulus** *size* **Example:**	Generates a RSA key pair for signing and encryption. You can also assign a label to each key pair using the label keyword. The label is referenced by the trustpoint that uses

	Command or Action	Purpose
	`Device(config)# crypto key generate rsa label general-keys  modulus 2048`	the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>. If you do not use additional keywords this command generates one general purpose RSA key pair. If the modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the modulus keyword.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show authentication session interface** *interface-id* **Example:** `Device# show authentication session interface gigabitethernet 0/1/1`	Verifies the authorized session security status.

Configuring Enrollment using SCEP

Simple Certificate Enrollment Protocol (SCEP) is a Cisco-developed enrollment protocol that uses HTTP to communicate with the certificate authority (CA) or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name* **Example:** `Device(config)# crypto pki trustpoint ka`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem* **Example:** `Device(ca-trustpoint)# enrollment url http://url:80`	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.

	Command or Action	Purpose
Step 5	**rsakeypair** *key-label key-sizeencryption-key-size* **Example:** `Device(ca-trustpoint)# ` **`rsakeypair exampleCAkeys`**	Specifies which key pair to associate with the certificate. - A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. - Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The *key-size* and *encryption-key-size* must be the same size. Length of less than 2048 is not recommended. **Note** The **rsakeypair** name must match the trust-point name. **Note** If this command is not enabled, the FQDN key pair is used.
Step 6	**serial-number none** **Example:** `Device(ca-trustpoint)# ` **`serial-number none`**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none** **Example:** `Device(ca-trustpoint)# ` **`ip-address none`**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl** **Example:** `Device(ca-trustpoint)# ` **`revocation-check crl`**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**auto-enroll** *percent* **regenerate** **Example:** `Device(ca-trustpoint)# ` **`auto-enroll 90 regenerate`**	Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA. If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration. By default, only the Domain Name System (DNS) name of the device is included in the certificate. Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will

	Command or Action	Purpose
		appear in the trustpoint configuration to indicate whether the key pair is exportable: “! RSA key pair associated with trustpoint is exportable.” It is recommended that a new key pair be generated for security reasons.
Step 10	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 11	**crypto pki authenticate** *name* **Example:** `Device(config)# crypto pki authenticate myca`	Retrieves the CA certificate and authenticates it.

Step 12	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 13	**show crypto pki certificate** *trustpoint name* **Example:** `Device# show crypto pki certificate ka`	Displays information about the certificate for the trust point.

Configuring Enrollment Manually

If your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform the following task to set up manual certificate enrollment:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name* **Example:** `Device# crypto pki trustpoint ka`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url-name* **Example:**	Specifies the URL of the CA on which your device should send certificate requests.

	Command or Action	Purpose
	`Device(ca-trustpoint)# enrollment url http://url:80`	An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**rsakeypair** *key-label key-sizeencryption-key-size* **Example:** `Device(ca-trustpoint)#  rsakeypair exampleCAkeys`	Specifies which key pair to associate with the certificate. • A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The *key-size* and *encryption-key-size* must be the same size. Length of less than 2048 is not recommended. **Note** The **rsakeypair** name must match the trust-point name. **Note** If this command is not enabled, the FQDN key pair is used.
Step 6	**serial-number none** **Example:** `Device(ca-trustpoint)# serial-number none`	Specifies that serial numbers will not be included in the certificate request.
Step 7	**ip-address none** **Example:** `Device(ca-trustpoint)# ip-address none`	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl** **Example:** `Device(ca-trustpoint)# revocation-check crl`	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 10	**crypto pki authenticate** *name* **Example:** `Device(config)# crypto pki authenticate myca`	Retrieves the CA certificate and authenticates it.
Step 11	**crypto pki enroll** *name* **Example:**	Generates certificate request and displays the request for copying and pasting into the certificate server.

	Command or Action	Purpose
	`Device(config)# crypto pki enroll myca`	Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate** **Example:** `Device(config)# crypto pki import myca certificate`	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.

Step 13	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 14	**show crypto pki certificate** *trustpoint name* **Example:** `Device# show crypto pki certificate  ka`	Displays information about the certificate for the trust point.

Enabling 802.1x Authentication and Configuring AAA

SUMMARY STEPS

1. enable
2. configure terminal
3. aaa new-model
4. dot1x system-auth-control
5. radius server *name*
6. address *ip-address* auth-port *port-number* acct-port *port-number*
7. automate-tester username username

8. key *string*
9. radius-server deadtime *minutes*
10. exit
11. aaa group server radius *group-name*
12. server *name*
13. exit
14. aaa authentication dot1x default group *group-name*
15. aaa authorization network default group *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	aaa new-model **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	dot1x system-auth-control **Example:** `Device(config)# dot1x system-auth-control`	Enables 802.1X on your device.
Step 5	radius server *name* **Example:** `Device(config)# radius server ISE`	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	address *ip-address* auth-port *port-number* acct-port *port-number* **Example:** `Device(config-radius-server)# address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646`	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 7	automate-tester username username **Example:** `Device(config-radius-server)# automate-tester username dummy`	Enables the automated testing feature for the RADIUS server. With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks for a RADIUS response from the server. A success message is not necessary - a failed authentication suffices, because it shows that the server is alive.

	Command or Action	Purpose
Step 8	key *string* **Example:** `Device(config-radius-server)# key dummy123`	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
Step 9	radius-server deadtime *minutes* **Example:** `Device(config-radius-server)# radius-server deadtime 2`	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
Step 10	exit **Example:** `Device(config-radius-server)# exit`	Returns to global configuration mode.
Step 11	aaa group server radius *group-name* **Example:** `Device(config)# aaa group server radius ISEGRP`	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
Step 12	server *name* **Example:** `Device(config-sg)# server name ISE`	Assigns the RADIUS server name.
Step 13	exit **Example:** `Device(config-sg)# exit`	Returns to global configuration mode.
Step 14	aaa authentication dot1x default group *group-name* **Example:** `Device(config)# aaa authentication dot1x default group ISEGRP`	Sets the default authentication server group for IEEE 802.1x.
Step 15	aaa authorization network default group *group-name* **Example:** `aaa authorization network default group ISEGRP`	Sets the network authorization default group.

Configuring EAP-TLS Profile and 802.1x Credentials

SUMMARY STEPS

1. enable
2. configure terminal
3. eap profile *profile-name*
4. method tls
5. pki-trustpoint *name*
6. exit
7. dot1x credentials *profile-name*

8. username *username*
9. pki-trustpoint *name*
10. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	eap profile *profile-name* **Example:** `Device(config)# eap profile EAPTLS-PROF-IOSCA`	Configures EAP profile and enters EAP profile configurationmode.
Step 4	method tls **Example:** `Device(config-eap-profile)# method tls`	Enables EAP-TLS method on the device.
Step 5	pki-trustpoint *name* **Example:** `Device(config-eap-profile)# pki-trustpoint POLESTAR-IOS-CA`	Sets the default PKI trustpoint.
Step 6	exit **Example:** `Device(config-eap-profile)# exit`	Returns to global configuration mode.
Step 7	dot1x credentials *profile-name* **Example:** `Device(config)# dot1x credentials EAPTLSCRED-IOSCA`	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	username *username* **Example:** `Device(config-dot1x-cred)# username asr1000@polestar.company.com`	Sets the authentication user ID.
Step 9	pki-trustpoint *name* **Example:** `Device(config-dot1x-cred)# pki-trustpoint POLESTAR-IOS-CA`	Sets the default PKI trustpoint.

	Command or Action	Purpose

Step 10	end
	Example:
	`Device(config-dot1x-cred)# end`
		Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MACsec MKA using certificate-based MACsec encryption to interfaces, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface *interface-id* **Example:** `Device(config)# interface gigabitethernet 2/9`	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	macsec **Example:** `Device(config-if)# macsec`	Enables MACsec on the interface.
Step 5	authentication periodic **Example:** `Device(config-if)# authentication periodic`	(Optional) Enables reauthentication for this port.
Step 6	authentication timer reauthenticate interval **Example:** `Device(config-if)# authentication timer reauthenticate interval`	(Optional) Sets the reauthentication interval.
Step 7	access-session host-mode multi-domain **Example:** `Device(config-if)# access-session host-mode multi-domain`	Allows hosts to gain access to the interface.
Step 8	**access-session closed** **Example:**	Prevents preauthentication access on the interface.

	Command or Action	Purpose
	`Device(config-if)# access-session closed`	
Step 9	access-session port-control auto **Example:** `Device(config-if)# access-session port-control auto`	Sets the authorization state of a port.
Step 10	dot1x pae both **Example:** `Device(config-if)# dot1x pae both`	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	dot1x credentials *profile* **Example:** `Device(config-if)# dot1x credentials EAPTLSCRED-IOSCA`	Assigns a 802.1x credentials profile to the interface.
Step 12	dot1x supplicant eap profile *name* **Example:** `Device(config-if)# dot1x supplicant eap profile EAPTLS-PROF-IOSCA`	Assigns the EAP-TLS profile to the interface.
Step 13	dot1x authenticator eap profile *name* **Example:** `Device(config-if)# dot1x authenticator eap profile EAPTLS-PROF-IOSCA`	Assigns the EAP profile to use during 802.1x authentication.
Step 14	service-policy type control subscriber *control-policy name* **Example:** `Device(config-if)# service-policy type control subscriber DOT1X_POLICY_RADIUS`	Applies a subscriber control policy to the interface.
Step 15	exit **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 16	show macsec interface interface-id **Example:** `Device# show macsec interface GigabitEthernet 2/9`	Displays MACsec details for the interface.
Step 17	**show access-session interface** *interface-id* **details** **Example:** `Device# show access-session interface GigabitEthernet 2/9 details`	Verifies successful dot1x authentication and authorization. This is the first thing to check. If dot1x authentication fails, then MKA will never start.
Step 18	**show mka session interface** *interface-id* **details** **Example:**	Displays detailed MKA session status.

	Command or Action	Purpose
	Device# **show mka session interface GigabitEthernet 2/9 details**	

Example: Switch-to-Switch Certificate Based MACsec

An example configuration of switch-to-switch certificate based MACsec is shown below.

```
configure terminal
aaa new-model
aaa local authentication default authorization default
!
!
aaa authentication dot1x default group radius local
aaa authorization exec default local
aaa authorization network default group radius local
aaa authorization auth-proxy default group radius
aaa authorization credential-download default local
aaa accounting identity default start-stop group radius
!
!
aaa attribute list MUSTS
 attribute type linksec-policy must-secure
!
aaa attribute list macsec-dot1-credentials
 attribute type linksec-policy must-secure
!
aaa attribute list MUSTS_CA
 attribute type linksec-policy must-secure
!
aaa attribute list SHOULDS_CA
 attribute type linksec-policy should-secure
!
aaa attribute list mkadt_CA
 attribute type linksec-policy must-secure
!
aaa session-id common

username MUST aaa attribute list MUSTS_CA
username MUSTS.mkadt.cisco.com

crypto pki trustpoint demo
 enrollment terminal
 serial-number
 fqdn MUSTS.mkadt.cisco.com
 subject-name cn=MUSTS.mkadt.cisco.com,OU=CSG Security,O=Cisco Systems,L=Bengaluru,ST=KA,C=IN

 subject-alt-name MUSTS.mkadt.cisco.com
 revocation-check none
 rsakeypair demo 2048
 hash sha256

eap profile EAP_P
  method tls
 pki-trustpoint demo

dot1x system-auth-control
dot1x credentials MUSTS-CA
 username MUST
 password 0 MUST_CA
!
dot1x credentials MUSTS
```

```
 username MUSTS.mkadt.cisco.comcrypto pki authenticate demo

crypto pki authenticate
crypto pki enroll demo
crypto pki import demo certificate

policy-map type control subscriber MUSTS_1
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x both
 event authentication-failure match-all
  10 class always do-until-failure
   10 terminate dot1x
   20 authentication-restart 10
 event authentication-success match-all
  10 class always do-until-failure
   10 activate service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE

interface GigabitEthernet2/9
 switchport mode access
 macsec
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae both
 dot1x authenticator eap profile EAP_P
 dot1x credentials MUSTS
 dot1x supplicant eap profile EAP_P
 service-policy type control subscriber MUSTS_1
```

The following example shows output of the **show mka sessions** command for Switch-to-Switch Certificate Based MACsec.

```
show mka sessions

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

==========================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server

Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN

==========================================================================================
Gi2/14          40ce.24b7.617d/0002  pol_1             NO                 YES

2               f8b7.e2e5.ad88/0002  1                 Secured
80690202D09A9801BE98FC89D5380098



show mka sessions interface GigabitEthernet2/14 detail

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

Local Tx-SCI............. 40ce.24b7.617d/0002
Interface MAC Address.... 40ce.24b7.617d
MKA Port Identifier...... 2
Interface Name........... GigabitEthernet2/14
Audit Session ID......... 6514030B000000998FEDD629
CAK Name (CKN)........... 80690202D09A9801BE98FC89D538009800000000000000000000000000000000
Member Identifier (MI)... 534A6ECFBBA318B6423E49EB
```

```
Message Number (MN)...... 166
EAP Role................. Authenticator
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... 534A6ECFBBA318B6423E49EB00000001 (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)
SAK Rekey Time........... 0s (SAK Rekey interval not applicable)

MKA Policy Name.......... pol_1
Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000002 (GCM-AES-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 Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ---------------------------------------------------------------------------------------
  96E534A06B405442034B846E  163         f8b7.e2e5.ad88/0002  0         YES            0

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

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ---------------------------------------------------------------------------------------

show access-session interface GigabitEthernet2/14 detail
            Interface:  GigabitEthernet2/14
               IIF-ID:  0x1398D40E
          MAC Address:  f8b7.e2e5.ad88
         IPv6 Address:  Unknown
         IPv4 Address:  Unknown
            User-Name:  MUST
               Status:  Authorized
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
      Session timeout:  1800s (local), Remaining: 1470s
       Timeout action:  Reauthenticate
    Common Session ID:  6514030B000000998FEDD629
      Acct Session ID:  0x0000000e
```

```
                Handle:  0x5900003a
        Current Policy:  MUSTS_1


Local Policies:
                 Service Template: DEFAULT_LINKSEC_POLICY_MUST_SECURE (priority 150)

Server Policies:
      Security Policy:  Must Secure
      Security Status:  Link Secured


Method status list:
       Method           State
        dot1x           Authc Success
     dot1xSup           Authc Success
```

Configuring MKA/MACsec for Port Channel

Configuring MKA/MACsec for Port Channel Using PSK

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **macsec**
4. **mka policy** *policy-name*
5. **mka pre-shared-key key-chain** *key-chain-name*
6. **channel-group** *channel-group-number* **mode {active | passive } | {on }**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode.
Step 3	**macsec**	Enables MACsec on the interface. Supports layer 2 and layer 3 port channels.
Step 4	**mka policy** *policy-name*	Configures an MKA policy.
Step 5	**mka pre-shared-key key-chain** *key-chain-name*	Configures an MKA pre-shared-key key-chain name. **Note** The MKA pre-shared key can be configured on either physical interface or sub-interfaces and not on both.
Step 6	**channel-group** *channel-group-number* **mode {active | passive } | {on }**	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. The port channel associated with this channel group is automatically

	Command or Action	Purpose
		created if the port channel does not already exist.For mode, select one of the following keywords: • **on** — Forces the port to channel without PAgP or LACP. In the on mode, an EtherChannel exists only when a port group in the on mode is connected to another port group in the on mode. • **active** — Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets. • **passive** — Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.
Step 7	**end**	Returns to privileged EXEC mode.

Configuring Port Channel Logical Interfaces for Layer 2 EtherChannels

To create a port channel interface for a Layer 2 EtherChannel, perform this task:

SUMMARY STEPS

1. **configure terminal**
2. **[no] interface port-channel** *channel-group-number*
3. **switchport**
4. **switchport mode {access | trunk }**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**[no] interface port-channel** *channel-group-number*	Creates the port channel interface. **Note** Use the no form of this command to delete the port channel interface.
Step 3	**switchport**	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 4	**switchport mode {access	trunk }**	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks.
Step 5	**end**	Returns to privileged EXEC mode.

Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *interface-id*
3. **no switchport**
4. **ip address** *ip-address subnet_mask*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface port-channel** *interface-id*	Enters interface configuration mode.
Step 3	**no switchport**	Switches an interface that is in Layer 2 mode into Layer 3 mode for Layer 3 configuration.
Step 4	**ip address** *ip-address subnet_mask*	Assigns an IP address and subnet mask to the EtherChannel.
Step 5	**end**	Returns to privileged EXEC mode.

Example: Configuring MACsec MKA for Port Channel using PSK

Etherchannel Mode — Static/On

The following is a sample configuration on Device 1 and Device 2 with EtherChannel Mode on.

```
key chain KC macsec
  key 1000
    cryptographic-algorithm aes-128-cmac
    key-string FC8F5B10557C192F03F60198413D7D45
    end

mka policy POLICY
  key-server priority 0
  macsec-cipher-suite gcm-aes-128
  confidentiality-offset 0
  end

interface Te1/0/1
  channel-group 2 mode on
  macsec
  mka policy POLICY
  mka pre-shared-key key-chain KC
  end

interface Te1/0/2
  channel-group 2 mode on
  macsec
  mka policy POLICY
```

```
   mka pre-shared-key key-chain KC
   end
```

Layer 2 EtherChannel Configuration

Device 1

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

The following shows a sample output of **show etherchannel summary** command.

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)          -        Te1/0/1(P)  Te1/0/2(P)
```

Layer 3 EtherChannel Configuration

Device 1

```
interface port-channel 2
 no switchport
 ip address 10.25.25.3 255.255.255.0
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 no switchport
 ip address 10.25.25.4 255.255.255.0
 no shutdown
 end
```

The following shows a sample output of **show etherchannel summary** command.

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-----------+----------+-------------------------------------------

 2      Po2(RU)          -        Te1/0/1(P)  Te1/0/2(P)
```

Etherchannel Mode — LACP

The following is a sample configuration on Device 1 and Device 2 with EtherChannel Mode as LACP.

```
 key chain KC macsec
   key 1000
     cryptographic-algorithm aes-128-cmac
     key-string FC8F5B10557C192F03F60198413D7D45
     end

 mka policy POLICY
   key-server priority 0
   macsec-cipher-suite gcm-aes-128
   confidentiality-offset 0
   end

 interface Te1/0/1
   channel-group 2 mode active
   macsec
   mka policy POLICY
   mka pre-shared-key key-chain KC
   end

 interface Te1/0/2
   channel-group 2 mode active
   macsec
   mka policy POLICY
   mka pre-shared-key key-chain KC
   end
```

Layer 2 EtherChannel Configuration

Device 1

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

The following shows a sample output of **show etherchannel summary** command.

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1


------+-------------+-----------+-------------------------------------------

 2      Po2(SU)         LACP      Te1/1/1(P)  Te1/1/2(P)
```

Layer 3 EtherChannel Configuration

Device 1

```
interface port-channel 2
 no switchport
 ip address 10.25.25.3 255.255.255.0
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 no switchport
 ip address 10.25.25.4 255.255.255.0
 no shut
```

The following shows a sample output of **show etherchannel summary** command.

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)         LACP      Te1/1/1(P)  Te1/1/2(P)
```

Displaying Active MKA Sessions

The following shows all the active MKA sessions.

```
# show mka sessions interface Te1/0/1
=========================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI         MACsec-Peers     Status            CKN

=========================================================================================
Te1/0/1        00a3.d144.3364/0025 POLICY           NO                NO

37             701f.539b.b0c6/0032 1                Secured
1000
```

Configuring MACsec Cipher Announcement

Configuring an MKA Policy for Secure Announcement

SUMMARY STEPS

1. **configure terminal**
2. **mka policy** *policy-name*
3. **key-server** *priority*
4. **[no] send-secure-announcements**
5. **macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}
6. **end**
7. **show mka policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mka policy** *policy-name*	Identify an MKA policy, and enter MKA policy configuration mode. The maximum policy name length is 16 characters. **Note** The default MACsec cipher suite in the MKA policy will always be "GCM-AES-128". If the device supports both "GCM-AES-128" and "GCM-AES-256" ciphers, it is highly recommended to define and use a user defined MKA policy to include both 128 and 256 bits ciphers or only 256 bits cipher, as may be required.
Step 3	**key-server** *priority*	Configure MKA key server options and set priority (between 0-255). **Note** When value of key server priority is set to 255, the peer can not become the key server. The key server priority value is valid only for MKA PSK; and not for MKA EAPTLS.
Step 4	**[no] send-secure-announcements**	Enables sending of secure announcements. Use the no form of the command to disable sending of secure announcements. By default, secure announcements are disabled.
Step 5	**macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}	Configures cipher suite for deriving SAK with 128-bit or 256-bit encryption.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show mka policy**	Verify your entries.

Configuring Secure Announcement Globally (Across all the MKA Policies)

SUMMARY STEPS

1. **configure terminal**
2. **[no] mka defaults policy send-secure-announcements**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**[no] mka defaults policy send-secure-announcements**	Enables sending of secure announcements in MKPDUs across MKA policies. By default, secure announcements are disabled.
Step 3	**end**	Returns to privileged EXEC mode.

Configuring EAPoL Announcements on an interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **[no] eapol annoucement**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 3	**[no] eapol annoucement**	Enable EAPoL announcements. Use the no form of the command to disable EAPoL announcements. By default,EAPoL announcements are disabled.
Step 4	**end**	Returns to privileged EXEC mode.

Examples: Configuring MACsec Cipher Announcement

This example shows how to configure MKA policy for Secure Announcement:

```
# configure terminal
(config)# mka policy mka_policy
(config-mka-policy)# key-server 2
```

```
(config-mka-policy)# send-secure-announcements
(config-mka-policy)#macsec-cipher-suite gcm-aes-128confidentiality-offset 0
(config-mka-policy)# end
```

This example shows how to configure Secure Announcement globally:

```
# configure terminal
(config)# mka defaults policy send-secure-announcements
(config)# end
```

This example shows how to configure EAPoL Announcements on an interface:

```
# configure terminal
(config)# interface GigabitEthernet 1/0/1
(config-if)# eapol announcement
(config-if)# end
```

The following is a sample output for **show running-config interface** *interface-name* command with EAPoL announcement enabled.

```
# show running-config interface GigabitEthernet 1/0/1
switchport mode access
 macsec
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae authenticator
 dot1x timeout quiet-period 10
 dot1x timeout tx-period 5
 dot1x timeout supp-timeout 10
 dot1x supplicant eap profile peap
 eapol announcement
 spanning-tree portfast
 service-policy type control subscriber Dot1X
```

The following is a sample output of the **show mka sessions interface** *interface-name* **detail** command with secure announcement disabled.

```
# show mka sessions interface GigabitEthernet 1/0/1 detail


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

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89567
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC
```

```
Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... D46CBEC05D5D67594543CEAE00000001 (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.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
Send Secure Announcement.. DISABLED
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 Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
  38046BA37D7DA77E06D006A9  89555       c800.8459.e764/002a   10

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

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
```

The following is a sample output of the **show mka sessions details** command with secure announcement disabled.

```
# show mka sessions details
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
```

```
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... D46CBEC05D5D67594543CEAE00000001 (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.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
Send Secure Announcement.. DISABLED
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 Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
  38046BA37D7DA77E06D006A9  89560       c800.8459.e764/002a   10

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

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority
  ----------------------------------------------------------------------
```

The following is a sample output of the **show mka policy** *policy-name* **detail** command with secure announcement disabled.

```
# show mka policy p2 detail
MKA Policy Configuration ("p2")
========================
MKA Policy Name........ p2
Key Server Priority.... 2
```

```
Confidentiality Offset. 0
Send Secure Announcement..DISABLED
Cipher Suite(s)........ GCM-AES-128

Applied Interfaces...
  GigabitEthernet1/0/1
```