



Configuring MACsec Encryption

- [Prerequisites for MACsec Encryption, on page 1](#)
- [Restrictions for MACsec Encryption, on page 2](#)
- [Information About MACsec Encryption, on page 3](#)
- [How to Configure MACsec Encryption, on page 9](#)
- [Configuring Examples for MACsec Encryption, on page 28](#)
- [Example: configuring host to switch MACsec, on page 44](#)
- [Example: configure multi-domain, on page 45](#)
- [Additional References for MACsec Encryption, on page 46](#)
- [eEdge Integration with MACsec, on page 47](#)

Prerequisites for MACsec Encryption

Prerequisites for MACsec Encryption

This section list the prerequisites for MACsec encryption:

- Enable the **ssci-based-on-sci** command while configuring MACsec encryption on the device to allow interoperability with non-Cisco and non-IOS XE devices.
- Ensure that 802.1x authentication and AAA are configured on your device.
- You must configure the **flowcontrol receive desired** command on all MACsec-enabled ports to enable flowcontrol explicitly.

Prerequisites for Certificate-Based MACsec

This section list the prerequisites for Certificate-Based MACsec:

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.

Restrictions for MACsec Encryption

- For IE3505 switches, MACsec is not supported with the FPGA features like HSR/PRP, DLR.
- The Network Essentials package supports MACsec with 128-bit encryption, while the Network Advantage package supports MACsec with 256-bit encryption.
- MACsec with MACsec Key Agreement (MKA) is supported only on point-to-point links.
- MACsec configuration is not supported on EtherChannel ports. Instead, MACsec configuration can be applied on the individual member ports of an EtherChannel. To remove MACsec configuration, you must first unbundle the member ports from the EtherChannel, and then remove it from the individual member ports.
- Host-to-switch MKA MACsec supports only 128-bit MACsec encryption for host to switch 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.
- Packet number exhaustion rekey is not supported.
- If the **dot1q tag vlan native** command is configured globally, the dot1x reauthentication will fail on trunk ports.
- MACsec XPN Cipher Suites do not provide confidentiality protection with a confidentiality offset, and these together are not supported in switch-to-switch MACsec connections.
- As per IEEE standards, the maximum value of replay window is 2^{30-1} for MACsec XPN Cipher Suites. Even if you configure a higher value than this, it will be restricted to 2^{30-1} only.
- MACsec with Precision Time Protocol (PTP) is not supported.
- MACsec switch-to-host connections in an overlay network are not supported.
- MACsec is not supported with Multicast VPN (mVPN).
- The **should-secure** access mode is supported on switch-to-switch ports only using PSK authentication.
- PSK fallback key chain is not supported on Ethernet Virtual Circuit (EVC) and point-to-multipoint cases.
- PSK fallback key chain supports infinite lifetime with one key only. The connectivity association key name (CKN) ID used in the fallback key chain must not match any of the CKN IDs used in the primary key chain.
- EAPOL packets of EtherType 0x888E are not intercepted by the interface if MACsec or dot1x is not enabled on the interface.
- If there are any intermediate switches present between two MACsec endpoints, any P2P protocols like STP or CDP will not work.
- Network-Based Application Recognition (NBAR) is not supported on MACsec switch-to-host connections.
- MACsec Cipher Announcement feature is not supported.
- Dot1x CTS with the Certificate based MACsec feature is not supported.
- Configuring custom EAPOL is not supported.

Information About MACsec Encryption

The following sections provide information about MACsec encryption.

Recommendations for MACsec Encryption

This section lists the recommendations for configuring MACsec encryption:

- Use the confidentiality (encryption) offset as 0 in switch-to-host connections.
- Use Bidirectional Forwarding and Detection (BFD) timer value as 750 milliseconds for 10Gbps ports and 1.25 seconds for any port with speed above 10Gbps.
- Execute the **shutdown** command, and then the **no shutdown** command on a port, after changing any MKA policy or MACsec configuration for active sessions, so that the changes are applied to active sessions.
- Set the connectivity association key (CAK) rekey overlap timer to 30 seconds or more.
- Do not use Cisco TrustSec Security Association Protocol (SAP) MACsec encryption for port speeds above 10Gbps.
- Do not enable both Cisco TrustSec SAP and uplink MKA at the same time on any interface.
- Use MACsec MKA encryption.
- Do not use the **delay-protection** command when defining MKA policy if MACsec scale sessions are configured.



Note Whenever any change is performed on MKA policy or MACsec configuration on interface, it is mandatory to perform shutdown on interface and ensure that interface is down and then perform no shutdown on interface.

MACsec Encryption Overview

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Switches support 802.1AE encryption with MACsec Key Agreement (MKA) 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 both Cisco TrustSec Network Device Admission Control (NDAC), Security Association Protocol (SAP) and MKA-based key exchange protocol.



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

Link layer security can include both packet authentication between switches and MACsec encryption between switches (encryption is optional). Link layer security is supported on SAP-based MACsec.

Table 1: MACsec Support on Switch Ports

Connections	MACsec support
Switch-to-switch	MACsec MKA encryption (recommended) Cisco TrustSec SAP

Cisco TrustSec and Cisco SAP are meant only for switch-to-switch links and are 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. Cisco NDAC and SAP are mutually exclusive with Network Edge Access Topology (NEAT), which is used for compact switches to extend security outside the wiring closet.

Media Access Control Security and MACsec Key Agreement

MACsec, defined in 802.1AE, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. MKA and MACsec are implemented after successful authentication using certificate-based MACsec or Pre Shared Key (PSK) framework.

A switch using MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the MKA peer. MACsec frames are encrypted and protected with an integrity check value (ICV). When the switch receives frames from the MKA peer, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The switch compares that ICV to the ICV within the frame. If they are not identical, the frame is dropped. The switch also encrypts and adds an ICV to any frames sent over the secured port (the access point used to provide the secure MAC service to a MKA peer) using the current session key.

The MKA Protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1x-REV. The MKA Protocol extends 802.1x to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers.

The EAP framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). The switch acts as the authenticator for both uplink and downlink; and acts as the key server for downlink. It generates a random secure association key (SAK), which is sent to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the switch sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes with no MKPDU received from a participant. For example, if a MKA peer disconnects, the participant on the switch continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the MKA peer.



Note Integrity check value (ICV) indicator in MKPDU is optional. ICV is not optional when the traffic is encrypted.

EAPoL Announcements indicate the use of the type of keying material. The announcements can be used to announce the capability of the supplicant as well as the authenticator. Based on the capability of each side, the largest common denominator of the keying material could be used.

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

Definition of Policy-Map Actions

This section describes the policy-map actions and its definition:

- Activate: Applies a service template to the session.
- Authenticate: Starts authentication of the session.
- Authorize: Explicitly authorizes a session.
- Set-domain: Explicitly sets the domain of a client.
- Terminate: Terminates the method that is running, and deletes all the method details associated with the session.
- Deactivate: Removes the service-template applied to the session. If not applied, no action is taken.
- Set-timer: Starts a timer and gets associated with the session. When the timer expires, any action that needs to be started can be processed.
- Authentication-restart: Restarts authentication.
- Clear-session: Deletes a session.
- Pause: Pauses authentication.

Rest of the actions as self-explanatory and are associated with authentication.

Virtual Ports

Use virtual ports for multiple secured connectivity associations on a single physical port. Each connectivity association (pair) represents a virtual port. In uplink, you can have only one virtual port per physical port. You cannot simultaneously host secured and unsecured sessions in the same VLAN on the same port. Because of this limitation, 802.1x multiple authentication mode is not supported.

The exception to this limitation is in multiple-host mode when the first MACsec supplicant is successfully authenticated and connected to a hub that is connected to the switch. A non-MACsec host connected to the hub can send traffic without authentication because it is in multiple-host mode. We do not recommend using multi-host mode because after the first successful client, authentication is not required for other clients.

Virtual ports represent an arbitrary identifier for a connectivity association and have no meaning outside the MKA Protocol. A virtual port corresponds to a separate logical port ID. Valid port IDs for a virtual port are 0x0002 to 0xFFFF. Each virtual port receives a unique secure channel identifier (SCI) based on the MAC address of the physical interface concatenated with a 16-bit port ID.

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. See [Example: Displaying MKA Information, on page 38](#) for further information.

Key Lifetime and Hitless Key Rollover

A MACsec key chain can have multiple pre-shared keys (PSK) each configured with a key id and an optional lifetime. A key lifetime specifies at which time the key expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the key chain after the lifetime is expired. Time zone of the key can be local or UTC. Default time zone is UTC.

You can Key rolls over to the next key within the same key chain by configuring a second key in the key chain and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless, that is, key rolls over without traffic interruption.

On all participating devices, the MACsec key chain must be synchronised by using Network Time Protocol (NTP) and the same time zone must be used. If all the participating devices are not synchronized, the connectivity association key (CAK) rekey will not be initiated on all the devices at the same time.



Note The lifetime of the keys need to be overlapped in order to achieve hitless key rollover.

Default timer-based SAK rekey

A Secure Association Key (SAK) is a core security feature in MACsec (Media Access Control Security) that:

- delivers encryption and authentication for Ethernet frames, ensuring secure data transmission, and
- is dynamically created and managed as secure associations are established between network devices.

Cisco Catalyst IE3500 Rugged Series Switches support default timer-based rekey, with rekey frequencies based on interface speed:

- 1 Gbps: approximately 30 minutes
- 10 Gbps: approximately 3 minutes
- 40 Gbps: approximately 45 seconds



Note The Cisco Catalyst IE3500 Rugged Series Switches doesn't support packet numbering (PN) exhaustion rekey.

Fallback Key

The Fallback Key feature establishes an MKA session with the pre-shared fallback key whenever the primary pre-shared key (PSK) fails to establish a session because of key mismatch. This feature prevents downtime and ensures traffic hitless scenario during CAK mismatch (primary PSK) between the peers. The purpose of

the fallback key chain is to act as a last resort key. The fallback key feature is only applicable for PSK based MKA or MACsec sessions.

Replay Protection Window Size

Replay protection is a feature provided by MACsec to counter replay attacks. Each encrypted packet is assigned a unique sequence number and the sequence is verified at the remote end. Frames transmitted through a Metro Ethernet service provider network are highly susceptible to reordering due to prioritization and load balancing mechanisms used within the network.

A replay window is necessary to support the use of MACsec over provider networks that reorder frames. Frames within the window can be received out of order, but are not replay protected. The default window size is 0, which enforces strict reception ordering. The replay window size can be configured in the range of 0 to $2^{32} - 1$.

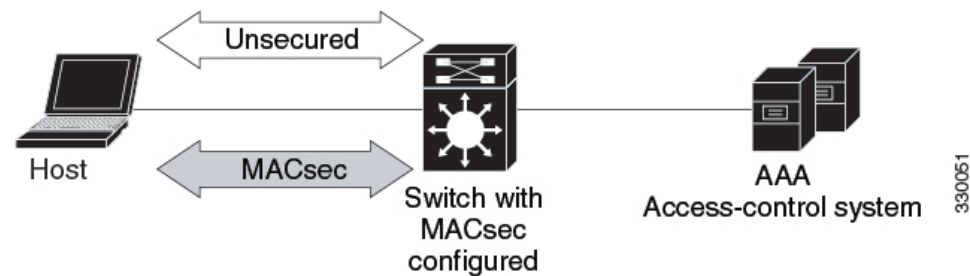
MACsec, MKA, and 802.1x Host Modes

You can use MACsec and the MKA Protocol with 802.1x single-host mode, multi-host mode, or Multi Domain Authentication (MDA) mode. Multiple authentication mode is not supported.

Single-Host Mode

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

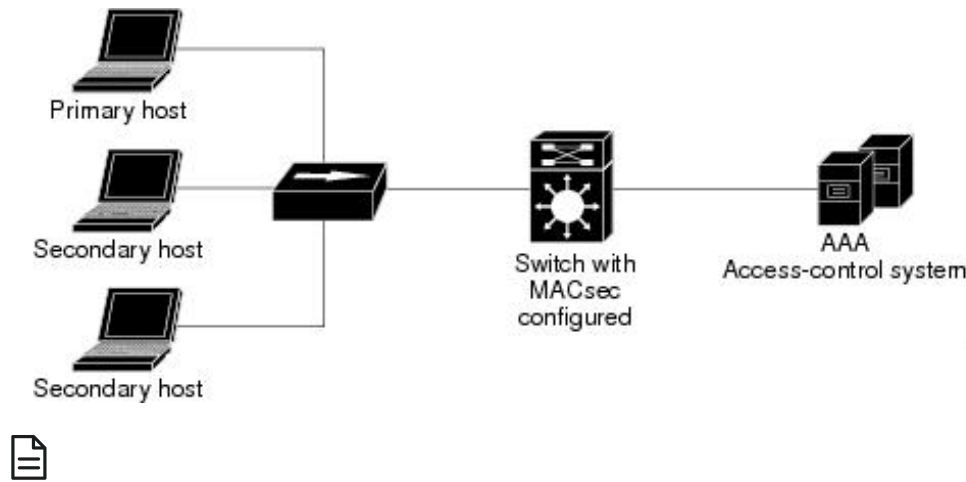
Figure 1: MACsec in Single-Host Mode with a Secured Data Session



Multiple Host Mode

In standard (not 802.1x REV) 802.1x multiple-host mode, a port is open or closed based on a single authentication. If one user, the primary secured client services client host, is authenticated, the same level of network access is provided to any host connected to the same port. If a secondary host is a MACsec supplicant, it cannot be authenticated and traffic would not flow. A secondary host that is a non-MACsec host can send traffic to the network without authentication because it is in multiple-host mode. The figure shows MACsec in Standard Multiple-Host Unsecure Mode.

Figure 2: MACsec in Multiple-Host Mode - Unsecured



Note Multi-host mode is not recommended because after the first successful client, authentication is not required for other clients, which is not secure.

Multiple-Domain Mode

In standard (not 802.1x REV) 802.1x multiple-domain mode, a port is open or closed based on a single authentication. If the primary user, a PC on data domain, is authenticated, the same level of network access is provided to any domain connected to the same port. If a secondary user is a MACsec supplicant, it cannot be authenticated and traffic would not flow. A secondary user, an IP phone on voice domain, that is a non-MACsec host, can send traffic to the network without authentication because it is in multiple-domain mode.

MACsec MKA using Certificate-based MACsec

MACsec MKA is supported on switch-to-switch links. Using certificate-based MACsec, you can configure MACsec MKA between device uplink ports. Certificate-based MACsec allows mutual authentication and obtains an MSK (master session key) from which the connectivity association key (CAK) is derived for MKA operations. Device certificates are carried, using certificate-based MACsec, for authentication to the AAA server.

Prerequisites for MACsec MKA using Certificate-based MACsec

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.
- Ensure that 802.1x authentication and AAA are configured on your device.

MACsec Connection Across Intermediate Switches

The encrypted packets were dropped if WAN MACsec was configured on the end devices with MACsec not configured on the intermediate switches. With the ClearTag feature implemented on the ASIC, the switch forwards the encrypted packet without parsing the MACsec header.

Limitations for MACsec Connections Across Intermediate Switches

- Hop-by-hop MACsec encryption with switches as intermediate switches where WAN MACsec is configured on the routers is not supported.
- WAN MACsec configured on the routers with intermediate switches as the switches is not supported on Layer 3 VPNs.
- WAN MACsec configured on the routers with intermediate switches as the switches show CDP neighbors only in should-secure mode.

Switch-to-Switch MKA MACsec Must Secure Policy

Must-secure support is enabled on both the ingress and the egress. Must-secure is supported for MKA and SAP. With must-secure enabled, only EAPOL traffic will not be encrypted. The rest of the traffic will be encrypted. Unencrypted packets are dropped.



Note Must-secure mode is enabled by default.

With should-secure enabled, if the peer is configured for MACsec, the data traffic is encrypted, otherwise it is sent in clear text.

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.

It is recommended that you enable MKA/MACsec on all the member ports for better security of the port channel.

How to Configure MACsec Encryption

The following sections provide information about the various tasks that comprise MACsec encryption.

Configuring MKA and MACsec

MACsec is disabled by default. No MKA policies are configured.

Configure an MKA Policy

Perform these steps to configure MACsec Key Agreement (MKA) policy.

Procedure

Step 1 Accessing and creating the MKA policy

- a) Use the **enable** command to enable privileged EXEC mode.

Example:

```
Device> enable
```

Enter your password if prompted.

- b) Use the **configure terminal** command to enter global configuration mode.

Example:

```
Device# configure terminal
```

- c) Use the **mka policy** *policy-name* command to identifies an MKA policy, and enters into the MKA policy configuration mode.

Example:

```
Device(config)# mka policy mka_policy
```

The maximum policy name length is 16 characters.

Note

The default MACsec cipher suite in the MKA policy will always be "GCM-AES-128".

Step 2 Configuring MKA policy parameters

- a) Use the **key-server** command to configure MKA key server options.

Example:

```
Device(config-mka-policy)# key-server priority 200
```

It sets 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.

- b) Use **sak-rekey interval** *time-interval* to configure the SAK rekey interval in seconds.

Example:

```
Device(config-mka-policy) # sak-rekey interval 50
```

This configuration is optional, with a range from 30 to 65,535.

By default, the SAK rekey interval is set automatically based on interface speed. Any user-defined value must be less than the default interval.

Use the **no** form of this command to stop the SAK rekey timer.

- c) Use the **include-icv-indicator** command to enable the ICV indicator in MKPDU.

Example:

```
Device(config-mka-policy) # include-icv-indicator
```

Use the **no** form of this command to disable the ICV indicator.

- d) **macsec-cipher-suite** command to configure a cipher suite for deriving SAK with 128-bit encryption.

Example:

```
Device(config-mka-policy) # macsec-cipher-suite gcm-aes-128
```

The syntax includes: **macsec-cipher-suite gcm-aes-128**

- e) Use the **confidentiality-offset** command to set the confidentiality (encryption) offset for each physical interface.

Example:

```
Device(config-mka-policy) # confidentiality-offset 0
```

Note

Offset Value can be 0, 30 or 50. If you are using Anyconnect on the client, it is recommended to use Offset 0.

- f) Use the **ssci-based-on-sci** command to compute Short Secure Channel Identifier (SSCI) value.

Example:

```
Device(config-mka-policy) # ssci-based-on-sci
```

(Optional) SSCI computation value is based on Secure Channel Identifier (SCI) value. The higher the SCI value, the lower is the SSCI value.

- g) Use the **end** command to exit MKA policy configuration mode and return to privileged EXEC mode.

Example:

```
Device(config-mka-policy) # end
```

- h) (Optional) Use the **show mka policy** command to display MKA policy configuration information.

Example:

```
Device# show mka policy
```

Configuring Switch-to-host MACsec Encryption

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

Procedure

	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 1/1	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	switchport access vlanvlan-id Example: Device (config-if)# switchport access vlan 613	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	access-session host-mode multi-domain Example: Device (config-if)# access-session 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 8	access-session port-control auto Example: Device (config-if)# access-session 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 9	authentication periodic Example:	(Optional) Enables or disables re-authentication for this port .

	Command or Action	Purpose
	Device(config-if) # authentication periodic	
Step 10	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 11	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 12	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 13	dot1x pae authenticator Example: Device(config-if) # dot1x pae authenticator	Configures the port as an 802.1x port access entity (PAE) authenticator.
Step 14	service-policy type control subscriber control-policy-name Example: Device(config-if) # service-policy type control subscriber DOT1X_POLICY_RADIUS	Applies a previously configured control policy.
Step 15	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 16	end Example: Device(config) # end	Exits interface configuration mode and returns to privileged EXEC mode. For more information on host-to-switch MACsec configuration examples, see Example: configuring host to switch MACsec, on page 44 .
Step 17	show authentication session interface interface-id Example:	Verifies the authorized session security status.

	Command or Action	Purpose
	Device# show authentication session interface GigabitEthernet 1/1	
Step 18	show mka sessions Example: Device# show mka sessions	Verifies the established MKA sessions.

Configuring MACsec MKA using PSK

Beginning in privileged EXEC mode, follow these steps to configure MACsec MKA policies using a Pre Shared Key (PSK).

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	key chain key-chain-name macsec Example: Device(config)# key chain keychain1 macsec	Configures a key chain and enters the key chain configuration mode.
Step 4	key hex-string Example: Device(config-key-chain)# key 1000	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. Note For 128-bit encryption, use any value between 1 and 32 hex digit key-string.
Step 5	cryptographic-algorithm {aes-128-cmac aes-256-cmac} Example: Device(config-key-chain)# cryptographic-algorithm aes-128-cmac	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 6	key-string { [0 6 7] pwd-string pwd-string} Example: Device(config-key-chain)# key-string 12345678901234567890123456789012	Sets the password for a key string. Only hex characters must be entered.

	Command or Action	Purpose
Step 7	lifetime local [<i>start timestamp {hh::mm::ss day month year}</i>] [duration <i>seconds</i> <i>end timestamp {hh::mm::ss day month year}</i>] Example: Device(config-key-chain) # lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016	Sets the lifetime of the pre shared key.
Step 8	end Example: Device(config-key-chain) # end	Exits key chain configuration mode and returns to privileged EXEC mode.

Configuring MACsec MKA on an Interface using PSK

Beginning in privileged EXEC mode, follow these steps to configure MACsec MKA policies on an interface using a Pre Shared Key (PSK).

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 <i>interface-id</i> Example: Device(config-if) # interface GigabitEthernet 1/1	Enters interface configuration mode.
Step 4	macsec network-link Example: Device(config-if) # macsec network-link	Enables MACsec on the interface.
Step 5	mka policy <i>policy-name</i> Example: Device(config-if) # mka policy mka_policy	Configures an MKA policy.
Step 6	mka pre-shared-key key-chain <i>key-chain name</i> [fallback key-chain <i>key-chain name</i>] Example:	Configures an MKA pre-shared-key key-chain name.

	Command or Action	Purpose
	<code>Device(config-if)# mka pre-shared-key key-chain key-chain-name</code>	
Step 7	macsec replay-protection window-size <i>frame number</i> Example: <code>Device(config-if)# macsec replay-protection window-size 10</code>	Sets the MACsec window size for replay protection.
Step 8	end Example: <code>Device(config-if)# end</code>	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

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 network-link configuration on each of the participating node using the **no macsec network-link** command
2. Configure the MKA policy on the interface on each of the participating node using the **mka policy policy-name** command.
3. Enable the new session on each of the participating node by using the **macsec network-link** command.

Configuring MACsec MKA using Certificate-based MACsec

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

- Configure Certificate Enrollment
 - Generate Key Pairs
 - Configure SCEP Enrollment
 - Configure Certificates Manually
- Configure an Authentication Policy
- Configure certificate-based MACsec Profiles and IEEE 802.1x Credentials
- Configure MKA MACsec using certificate-based MACsec on Interfaces

Generating Key Pairs

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	Enter your password, if prompted.
Step 2	crypto key generate rsa label <i>label-name</i> general-keys modulus <i>size</i> Example: Device> crypto key generate rsa label general-keys modulus 2048	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 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 3	show authentication session interface <i>interface-id</i> Example: Device# show authentication session interface gigabitethernet 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 <i>server name</i> Example: Device(config)# crypto pki trustpoint ka	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	enrollment url <i>url name pem</i> Example: <pre>Device(ca-trustpoint) # enrollment url http://url:80</pre>	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.
Step 5	rsakeypair <i>label</i> Example: <pre>Device(ca-trustpoint) # rsakeypair exampleCAkeys</pre>	Specifies which key pair to associate with the certificate. Note The rsakeypair name must match the trust-point name.
Step 6	serial-number <i>none</i> Example: <pre>Device(ca-trustpoint) # serial-number none</pre>	The none keyword specifies that a serial number will not be included in the certificate request.
Step 7	ip-address <i>none</i> Example: <pre>Device(ca-trustpoint) # ip-address none</pre>	The none keyword specifies that no IP address should be included in the certificate request.
Step 8	revocation-check <i>crl</i> Example: <pre>Device(ca-trustpoint) # revocation-check crl</pre>	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	auto-enroll <i>percent regenerate</i> Example: <pre>Device(ca-trustpoint) # auto-enroll 90 regenerate</pre>	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.

	Command or Action	Purpose
		If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will 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:	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
	Device# crypto pki trustpoint ka	
Step 4	enrollment url <i>url name pem</i> 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.
Step 5	rsakeypair <i>label</i> Example: Device (ca-trustpoint) # rsakeypair exampleCAkeys	Specifies which key pair to associate with the certificate.
Step 6	serial-number <i>none</i> 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 <i>none</i> 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 <i>crl</i> 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 <i>name</i> Example: Device (config) # crypto pki authenticate myca	Retrieves the CA certificate and authenticates it.
Step 11	crypto pki enroll <i>name</i> Example: Device (config) # crypto pki enroll myca	Generates certificate request and displays the request for copying and pasting into the certificate server. Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request.

	Command or Action	Purpose
		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 <i>name</i> certificate Example: <pre>Device(config)# crypto pki import myca certificate</pre>	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: <pre>Device(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 14	show crypto pki certificate <i>trustpoint name</i> Example: <pre>Device# show crypto pki certificate ka</pre>	Displays information about the certificate for the trust point.

Configuring switch-to-switch MACsec encryption

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

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Device (config)# interface gigabitethernet 1/1	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	macsec network-link Example: Device (config-if)# macsec network-link	Enables MACsec on the interface.
Step 5	authentication periodic Example: Device (config-if)# authentication periodic	Enables reauthentication for this port.
Step 6	authentication timer reauthenticate interval Example: Device (config-if)# authentication timer reauthenticate interval	Sets the reauthentication interval.
Step 7	authentication host-mode multi-host Example: Device (config-if)# authentication host-mode multi-host	Allows hosts to gain access to the interface.
Step 8	access-session closed Example: Device (config-if)# access-session closed	Prevents preauthentication access on the interface.
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	service-policy type control subscriber control-policy-name Example: Device (config-if)# service-policy type control subscriber DOT1X_POLICY_RADIUS	Applies a previously configured control policy.

	Command or Action	Purpose
Step 12	dot1x credentials profile Example: Device(config-if)# dot1x credentials profile	Assigns a 802.1x credentials profile to the interface.
Step 13	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode. For more information on switch-to-switch MACsec configuration examples, see Example: configure multi-domain, on page 45 .
Step 14	show macsec interface interface-id Example: Device# show macsec interface GigabitEthernet 1/1	Displays MACsec details for the interface.

Configuring MKA/MACsec for Port Channel using PSK

Beginning in privileged EXEC mode, follow these steps to configure MKA policies on an interface using a Pre Shared Key (PSK).

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-if)# interface gigabitethernet 1/1	Enters interface configuration mode.
Step 4	macsec network-link Example: Device(config-if)# macsec network-link	Enables MACsec on the interface. Supports layer 2 and layer 3 port channels.
Step 5	mka policy policy-name Example: Device(config-if)# mka policy mka_policy	Configures an MKA policy.

	Command or Action	Purpose
Step 6	mka pre-shared-key key-chain <i>key-chain name</i> [fallback key-chain <i>key-chain name</i>] Example: <pre>Device(config-if)# mka pre-shared-key key-chain key-chain-name</pre>	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 7	macsec replay-protection window-size <i>frame number</i> Example: <pre>Device(config-if)# macsec replay-protection window-size 0</pre>	Sets the MACsec window size for replay protection.
Step 8	channel-group <i>channel-group-number</i> mode { active passive } Example: <pre>Device(config-if)# channel-group 3 mode active</pre>	Configures the port in a channel group and sets the mode. Note You cannot configure ports in a channel group without configuring MACsec on the interface. You must configure the commands in Step 3, 4, 5 and 6 before this step. The channel-number range is from 1 to 4096. The port channel associated with this channel group is automatically created if the port channel does not already exist. For mode, select one of the following keywords: • 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 9	end Example: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and 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:

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 port-channel <i>channel-group-number</i> Example: Device(config)# interface port-channel 1	Creates the port channel interface. Note Use the no form of this command to delete the port channel interface.
Step 4	switchport Example: Device(config-if)# switchport	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 5	switchport mode {access trunk} Example: Device(config-if)# switchport mode access	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks.
Step 6	end Example: Device(config-if)# end	Exits interface configuration mode and 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:

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.

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: Device (config) # interface gigabitethernet 1/1	Enters interface configuration mode.
Step 4	no switchport Example: Device (config-if) # no switchport	Switches an interface that is in Layer 2 mode into Layer 3 mode for Layer 3 configuration.
Step 5	ip address <i>ip-address subnet_mask</i> Example: Device (config-if) # ip address 192.0.2.10 255.255.255.254	Assigns an IP address and subnet mask to the EtherChannel.
Step 6	end Example: Device (config-if) # end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Cisco TrustSec MACsec

Configuring Cisco TrustSec Switch-to-Switch Link Security in Manual Mode

Before you begin

When manually configuring Cisco TrustSec on an interface, consider these usage guidelines and restrictions:

- If no SAP parameters are defined, Cisco TrustSec encapsulation or encryption is not performed.
- If you select GCM as the SAP operating mode, you must have a MACsec Encryption software license from Cisco. If you select GCM without the required license, the interface is forced to a link-down state.
- These protection levels are supported when you configure SAP pairwise master key (sap pmk):
 - SAP is not configured: no protection.
 - **sap mode-list gcm-encrypt gmac no-encap**: protection desirable but not mandatory.
 - **sap mode-list gcm-encrypt gmac**: confidentiality preferred and integrity required. The protection is selected by the supplicant according to supplicant preference.
 - **sap mode-list gmac**: integrity only.
 - **sap mode-list gcm-encrypt**: confidentiality required.
 - **sap mode-list gmac gcm-encrypt**: integrity required and preferred, confidentiality optional.
- Before changing the configuration from MKA to Cisco TrustSec SAP and vice versa, we recommend that you remove the interface configuration.

Beginning in privileged EXEC mode, follow these steps to manually configure Cisco TrustSec on an interface to another Cisco TrustSec device:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface interface-id Example: Device(config)# interface gigabitethernet 1/1	Note Enters interface configuration mode.
Step 3	cts manual Example: Device(config-if)# cts manual	Enters Cisco TrustSec manual configuration mode.
Step 4	sap pmk key [mode-list mode1 [mode2 [mode3 [mode4]]]] Example: Device(config-if-cts-manual)# sap pmk 1234abcdef mode-list gcm-encrypt null no-encap	(Optional) Configures the SAP pairwise master key (PMK) and operation mode. SAP is disabled by default in Cisco TrustSec manual mode. • key: A hexadecimal value with an even number of characters and a maximum length of 32 characters. The SAP operation mode options: • gcm-encrypt: Authentication and encryption Note Select this mode for MACsec authentication and encryption if your software license supports MACsec encryption. • gmac: Authentication, no encryption • no-encap: No encapsulation • null: Encapsulation, no authentication or encryption Note If the interface is not capable of data link encryption, no-encap is the default and the only available SAP operating mode. SGT is not supported.

	Command or Action	Purpose
Step 5	no propagate sgt Example: Device(config-if-cts-manual) # no propagate sgt	Use the no form of this command when the peer is incapable of processing a SGT. The no propagate sgt command prevents the interface from transmitting the SGT to the peer.
Step 6	exit Example: Device(config-if-cts-manual) # exit	Exits Cisco TrustSec 802.1x interface configuration mode.
Step 7	end Example: Device(config-if) # end	Returns to privileged EXEC mode.
Step 8	show cts interface [<i>interface-id</i> brief summary]	(Optional) Verify the configuration by displaying TrustSec-related interface characteristics.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Examples for MACsec Encryption

Example: Configuring MKA and MACsec

This example shows how to create an MKA policy:

```
Device> enable
Device# configure terminal
Device(config)# mka policy mka_policy
Device(config-mka-policy)# key-server priority 200
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 30
Device(config-mka-policy)# ssci-based-on-sci
Device(config-mka-policy)#end
```

Examples: Configuring MACsec MKA using PSK

This example shows how to configure MACsec MKA using PSK.

```
Device> enable
Device# configure terminal
Device(config)# Key chain keychain1 macsec
Device(config-keychain)# key 1000
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789012
```

```
Device(config-keychain-key)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016
Device(config-keychain-key)# end
```

This example shows how to configure MACsec MKA on an interface using PSK.

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

Examples: Configuring MACsec MKA using Certificate-based MACsec Remote Authentication

This example shows how to configure MACsec MKA using certificate-based MACsec remote authentication:

```
Device> enable
Device# configure terminal
Device(config)#aaa new-model
Device(config)#dot1x system-auth-control
Device(config)#radius server ISE
Device(config)#address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646
Device(config)#automate-tester username cisco
Device(config)#key cisco123
Device(config)#radius-server deadtime 2
!
Device(config)#aaa group server radius ISEGRP
Device(config)#server name ISE
!
Device(config)#aaa authentication dot1x default group ISEGRP
Device(config)#aaa authorization network default group ISEGRP
Device(config)#policy-map type control subscriber DOT1X_POLICY_RADIUS
Device(config)#event session-started match-all
Device(config)#10 class always do-until-failure
Device(config)#10 authenticate using dot1x both
Device(config)#event authentication-failure match-all
Device(config)#10 class always do-until-failure
Device(config)#10 terminate dot1x
Device(config)#20 authentication-restart 10
Device(config)#eap profile EAPTLS-PROF-IOSCA
Device(config)#method tls
Device(config)#pki-trustpoint POLESTAR-IO-CA
!
Device(config)#dot1x credentials EAPTLSCRED-IOSCA
Device(config)#username xyz@polestar.cisco.com
Device(config)#pki-trustpoint POLESTAR-IO-CA
Device(config-if)#interface GigabitEthernet1/1
Device(config-if)#macsec network-link
Device(config-if)#authentication periodic
Device(config-if)#authentication timer reauthenticate <reauthentication interval>
Device(config-if)#access-session host-mode multi-host
Device(config-if)#access-session closed
Device(config-if)#access-session port-control auto
Device(config-if)#dot1x pae both
Device(config-if)#dot1x credentials EAPTLSCRED-IOSCA
Device(config-if)#dot1x supplicant eap profile EAPTLS-PROF-IOSCA
Device(config-if)#service-policy type control subscriber DOT1X_POLICY_RADIUS
Device(config-if)# end
```

Examples: Configuring MACsec MKA using Certificate-based MACsec Local Authentication

Etherchannel Mode — Static/On

This example shows how to configure MACsec MKA using certificate-based MACsec remote authentication:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# dot1x system-auth-control
!
Device(config)# aaa authentication dot1x default local
Device(config)#aaa authorization network default local
Device(config)#aaa authorization credential-download default local
Device(config)#aaa local authentication default authorization default
Device(config)#aaa attribute list MUSTS-CA
Device(config)#attribute type linksec-policy must-secure
Device(config)#username MUST aaa attribute list MUSTS-CA
Device(config)#dot1x credentials MUSTS
Device(config)#username MUST
Device(config)#policy-map type control subscriber DOT1X_POLICY_LOCAL
Device(config)#event session-started match-all
Device(config)#10 class always do-until-failure
Device(config)#10 authenticate using dot1x both
Device(config)#event authentication-failure match-all
Device(config)#10 class always do-until-failure
Device(config)#10 terminate dot1x
Device(config)#20 authentication-restart 10
Device(config)#eap profile EAPTLS-PROF-IOSCA
Device(config)#method tls
Device(config)#pki-trustpoint POLESTAR-IOS-CA
!
Device(config-if)#interface GigabitEthernet1/1
Device(config-if)#macsec network-link
Device(config-if)#authentication periodic
Device(config-if)#authentication timer reauthenticate <reauthentication interval>
Device(config-if)#access-session host-mode multi-host
Device(config-if)#access-session closed
Device(config-if)#access-session port-control auto
Device(config-if)#dot1x pae both
Device(config-if)#dot1x credentials MUSTS
Device(config-if)#dot1x authenticator eap profile EAPTLS-PROF-IOSCA
Device(config-if)#dot1x supplicant eap profile EAPTLS-PROF-IOSCA
Device(config-if)#service-policy type control subscriber DOT1X_POLICY_RADIUS
Device(config-if)# end
```

Examples: Configuring MACsec MKA using Certificate-based MACsec Fallback Local Authentication

This example shows how to configure MACsec MKA using certificate-based MACsec fallback local authentication:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# dot1x system-auth-control
Device(config)# radius server ISE
```

```

Device(config)# address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646
Device(config)# automate-tester username cisco
Device(config)# key clsco123
Device(config)# radius-server deadtime 2
!
Device(config)#aaa group server radius ISEGRP
Device(config)#server name ISE
!
Device(config)#aaa authorization credential-download default group ISEGRP local
Device(config)#aaa local authentication default authorization default
Device(config)#aaa attribute list MUSTS-CA
Device(config)#attribute type linksec-policy must-secure
Device(config)#username MUST aaa attribute list MUSTS-CA
Device(config)#dot1x credentials MUSTS
Device(config)#username MUST
Device(config)#aaa authorization network default group ISEGRP local
Device(config)#policy-map type control subscriber DOT1X_POLICY_RADIUS
Device(config)#event session-started match-all
Device(config)#10 class always do-until-failure
Device(config)#10 authenticate using dot1x both
Device(config)#event authentication-failure match-all
Device(config)#10 class always do-until-failure
Device(config)#10 terminate dot1x
Device(config)#20 authentication-restart 10
Device(config)#eap profile EAPTLS-PROF-IOSCA
Device(config)#method tls
Device(config)#pki-trustpoint POLESTAR-IOS-CA
!
Device(config-if)#interface GigabitEthernet1/1
Device(config-if)#macsec network-link
Device(config-if)#authentication periodic
Device(config-if)#authentication timer reauthenticate <reauthentication interval>
Device(config-if)#access-session host-mode multi-host
Device(config-if)#access-session closed
Device(config-if)#access-session port-control auto
Device(config-if)#dot1x pae both
Device(config-if)#dot1x credentials MUSTS
Device(config-if)#dot1x authenticator eap profile EAPTLS-PROF-IOSCA
Device(config-if)#dot1x supplicant eap profile EAPTLS-PROF-IOSCA
Device(config-if)#service-policy type control subscriber DOT1X_POLICY_RADIUS
Device(config-if)# end

```

Example: Configuring MACsec MKA for Port Channel using PSK

Etherchannel Mode — Static/On

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

```

Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit

```

Layer 2 EtherChannel Configuration

Device 2

The following is sample output from the **show etherchannel summary** command:

A - formed by Auto LAG

Layer 3 EtherChannel Configuration

Device 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.10 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

Device 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.11 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

The following is sample output from the **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)	-	ge1/1 (P)	ge1/2 (P)
---	----------	---	-----------	-----------

Etherchannel Mode — LACP

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

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
```

Example: Configuring MACsec MKA for Port Channel using PSK

```

Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/1
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/1
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end

```

Layer 2 EtherChannel Configuration**Device 1**

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

Device 2

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

The following is sample output from the **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

```

Layer 3 EtherChannel Configuration

Device 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.12 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

Device 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.13 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

The following is sample output from the **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	qe1/1 (P)	qe1/2 (P)
---	----------	------	-----------	-----------

Etherchannel Mode — PAgP

The following is sample configuration on Device 1 and Device 2 with EtherChannel Mode as PAgP:

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
```

Example: Configuring MACsec MKA for Port Channel using PSK

```

Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/1
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/1
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end

```

Layer 2 EtherChannel Configuration**Device 1**

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

Device 2

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

The following shows a sample output from the **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)      PAgP      Te1/1 (P)  Te1/1 (P)
```

Layer 3 EtherChannel Configuration

Device 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.12 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

Device 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 192.0.2.13 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

The following is sample output from the **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)      PAgP      ge1/1 (P)  ge1/2 (P)
```

Displaying Active MKA Sessions

The following shows all the active MKA sessions.

Interface	Local-TxSCI	Policy-Name	Inherited	
Key-Server	Peer-RxSCI	MACsec-Peers	Status	CKN
ge1/1	00a3.d144.3364/0025	POLICY	NO	NO
37 1000	701f.539b.b0c6/0032	1	Secured	

The following is a sample output from the **show mka sessions** command:

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

[illegible]

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

[illegible]

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..... GigabitEthernet1/1  
Audit Session ID.....  
CAK Name (CKN)..... 0100000000000000000000000000000000000000000000000000000000000000  
Member Identifier (MI)... D46CBEC05D5D67594543CEAE  
Message Number (MN)..... 89567
```

Live Peers List:

Potential Peers List:

Dormant Peers List:

The following is a sample output from the **show mka sessions details** command:

```
Device# show mka sessions details
```

MKA Detailed Status for MKA Session

=====

Status: SECURED - Secured MKA Session with MACsec

Configuring MACsec Encryption

Example: Displaying MKA Information

```

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 from the **show mka policy** command:

```
Device# show mka policy
```

```
MKA Policy Summary...
```

Policy Name	KS Priority	Delay Protect	Replay Protect	Window Size	Conf Offset	Cipher Suite(s)	Interfaces 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	Gil/1

The following is a sample output from the **show mka policy policy-name** command:

```
Device# show mka policy p2
```

```
MKA Policy Summary...
```

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

The following is a sample output from the **show mka policy policy-name detail** command:

```
Device# show mka policy p2 detail
```

```
MKA Policy Configuration ("p2")
```


Example: Displaying MKA Information

```

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

```

The following is a sample output from the **show macsec interface** command:

```
Device# show macsec interface Gi 1/1
```

```

MACsec is enabled
Replay protect : enabled
Replay window : 0
Include SCI : yes
Use ES Enable : no

```

```
Use SCB Enable : no
Admin Pt2Pt MAC : forceTrue(1)
Pt2Pt MAC Operational : no
Cipher : GCM-AES-128
Confidentiality Offset : 0

Capabilities
ICV length : 16
Data length change supported: yes
Max. Rx SA : 16
Max. Tx SA : 16
Max. Rx SC : 8
Max. Tx SC : 8
Validate Frames : strict
PN threshold notification support : Yes
Ciphers supported : GCM-AES-128
                   GCM-AES-256
                   GCM-AES-XPB-128
                   GCM-AES-XPB-256

Access control : must secure

Transmit Secure Channels
SCI : 3C5731BBB5850475
SC state : inUse(1)
Elapsed time : 7w0d
Start time : 7w0d
Current AN: 0
Previous AN: -
Next PN: 149757
SA State: inUse(1)
Confidentiality : yes
SAK Unchanged : yes
SA Create time : 00:04:41
SA Start time : 7w0d
SC Statistics
Auth-only Pkts : 0
Auth-only Bytes : 0
Encrypted Pkts : 0
Encrypted Bytes : 0
SA Statistics
Auth-only Pkts : 0
Auth-only Bytes : 0
Encrypted Pkts : 149756
Encrypted Bytes : 16595088

Port Statistics
Egress untag pkts  0
Egress long pkts  0

Receive Secure Channels
SCI : 3C5731BBB5C504DF
SC state : inUse(1)
Elapsed time : 7w0d
Start time : 7w0d
Current AN: 0
Previous AN: -
Next PN: 149786
RX SA Count: 0
SA State: inUse(1)
SAK Unchanged : yes
SA Create time : 00:04:39
SA Start time : 7w0d
SC Statistics
```

```

Notvalid pkts 0
Invalid pkts 0
Valid pkts 0
Late pkts 0
Uncheck pkts 0
Delay pkts 0
UnusedSA pkts 0
NousingSA pkts 0
Validated Bytes 0
Decrypted Bytes 0
SA Statistics
Notvalid pkts 0
Invalid pkts 0
Valid pkts 149784
Late pkts 0
Uncheck pkts 0
Delay pkts 0
UnusedSA pkts 0
NousingSA pkts 0
Validated Bytes 0
Decrypted Bytes 16654544

Port Statistics
Ingress untag pkts 0
Ingress notag pkts 631726
Ingress badtag pkts 0
Ingress unknownSCI pkts 0
Ingress noSCI pkts 0
Ingress overrun pkts 0

```

Example: configuring host to switch MACsec

This example shows how to configure host to switch MACsec:

```

Device>enable
Device#configure terminal
Device(config)#aaa new-model
!
Device(config)#aaa group server radius tests
Device(config)#server name RAD-1
!
Device(config)#aaa authentication dot1x default group tests
Device(config)#aaa authorization network default group tests
!
Device(config)#dot1x system-auth-control
!
Device(config)#policy-map type control subscriber TEST
Device(config)#event session-started match-all
Device(config)#10 class always do-until-failure
Device(config)#10 authenticate using dot1x priority 10
!
Device(config-if)#interface GigabitEthernet1/2
Device(config-if)#macsec
Device(config-if)#switchport access vlan 613
Device(config-if)#switchport mode access
Device(config-if)#access-session host-mode single-host
Device(config-if)#access-session port-control auto
Device(config-if)#dot1x pae authenticator
Device(config-if)#service-policy type control subscriber TEST
!
Device(config-if)#radius server RAD-1

```

```
Device(config-if)#address ipv4 <ISE ipv4 address> auth-port 1812 acct-port 1813
Device(config-if)#key cisco
```

Example: configure multi-domain

This example shows how to configure multi-domain MACsec:

```
Device>enable
Device#configure terminal
Device(config)#aaa new-model
!
Device(config)#aaa group server radius tests
Device(config)#server name RAD-1
!
Device(config)#aaa authentication dot1x default group tests
Device(config)#aaa authorization network default group tests
!
Device(config)#aaa server radius dynamic-author
Device(config)#client <ipv4 address> server-key cisco
!

Device(config)#dot1x system-auth-control
!

Device(config)#class-map type control subscriber match-all DOT1X

Device(config)#match method dot1x
!
Device(config)#class-map type control subscriber match-all DOT1X_FAILED

Device(config)#match method dot1x
Device(config)#match result-type method dot1x authoritative
!
Device(config)#class-map type control subscriber match-all DOT1X_NO_RESP
Device(config)#match method dot1x
Device(config)#match result-type method dot1x agent-not-found
!
Device(config)#class-map type control subscriber match-all MAB
Device(config)#match method mab
!
Device(config)#class-map type control subscriber match-all MAB_FAILED
Device(config)#match method mab
Device(config)#match result-type method mab authoritative
!
Device(config)#policy-map type control subscriber TEST4
Device(config)#event session-started match-all
Device(config)#10 class always do-until-failure
Device(config)#10 authenticate using dot1x priority 10
Device(config)#20 authenticate using mab priority 20
Device(config)#Device(config)#event authentication-failure match-first
Device(config)#10 class DOT1X_FAILED do-until-failure
Device(config)#10 terminate dot1x
Device(config)#20 class MAB_FAILED do-until-failure
Device(config)#10 terminate mab
Device(config)#20 authenticate using dot1x priority 10
Device(config)#30 class DOT1X_NO_RESP do-until-failure
Device(config)#10 terminate dot1x
Device(config)#20 authentication-restart 60
Device(config)#40 class always do-until-failure
Device(config)#10 terminate mab
Device(config)#20 terminate dot1x
```

```

Device(config)#30 authentication-restart 60
Device(config)#event agent-found match-all
Device(config)#10 class always do-until-failure
Device(config)#10 terminate mab
Device(config)#20 authenticate using dot1x priority 10
Device(config)#event authentication-success match-all
Device(config)#10 class always do-until-failure
Device(config)#10 activate service-template DEFAULT_LINKSEC_POLICY_SHOULD_SECURE
!
Device(config-if)#interface GigabitEthernet1/2
Device(config-if)#macsec
Device(config-if)#switchport access vlan 613
Device(config-if)#switchport mode access
Device(config-if)#switchport voice vlan 612
Device(config-if)#access-session host-mode multi-domain
Device(config-if)#access-session port-control auto
Device(config-if)#mab
Device(config-if)#dot1x pae authenticator
Device(config-if)#spanning-tree portfast
Device(config-if)#service-policy type control subscriber TEST4
!
Device(config)#radius-server attribute 6 on-for-login-auth
Device(config)#radius-server attribute 8 include-in-access-req
Device(config)#radius-server attribute 25 access-request include
Device(config)#radius-server vsa send cisco-nas-port
!
Device(config)#radius server RAD-1
Device(config)#address ipv4 <ISE ipv4 address> auth-port 1812 acct-port 1813
Device(config)#key cisco

```

Additional References for MACsec Encryption

Standards and RFCs

Standard/RFC	Title
IEEE 802.1AE-2006	<i>Media Access Control (MAC) Security</i>
IEEE 802.1X-2010	<i>Port-Based Network Access Control</i>
IEEE 802.1AEbw-2013	<i>Media Access Control (MAC) Security (Amendment to IEEE 802.1AE-2006)—Extended Packet Numbering (XPN)</i>
IEEE 802.1Xbx-2014	<i>Port-Based Network Access Control (Amendment to IEEE 802.1X-2010)</i>
RFC 4493	<i>The AES-CMAC Algorithm</i>

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

eEdge Integration with MACsec

The Media Access Control Security (MACsec) standard is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The eEdge Integration with MACsec feature allows you to integrate the MACsec standard with enterprise edge (eEdge) devices to enhance Session Aware Networking capabilities. Session Aware Networking provides a policy and identity-based framework for edge devices to deliver flexible and scalable services to subscribers.

Prerequisites for eEdge Integration with MACsec

- Layer 2 encryption protocols like the IEEE 802.1AE Media Access Control Security (MACsec) standard must register with the eEdge session manager to receive disconnect notifications and perform cleanup.
- You must provision one virtual interface per secure association.

Restrictions for eEdge Integration with MACsec

- The Media Access Control Security (MACsec) standard is supported only in single-host and multihost modes. If a link layer security policy is configured as must-secure and the host mode is not configured as a single host or a multihost, the connection is closed.
- The MACsec standard is not supported in multi-authentication mode.
- The MACsec standard supports the 802.1AE encryption with MACsec Key Agreement (MKA) only on downlink ports for encryption between a MACsec-capable device and host devices.

Information About eEdge Integration with MACsec

The following sections provide information about eEdge Integration with MACsec feature.

Overview of MACsec

Media Access Control Security (MACsec) is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Implementing the MACsec encryption standard enables support for the 802.1AE encryption with MACsec Key Agreement (MKA) on downlink ports for encryption between a MACsec-capable device and host devices. The MACsec-capable device also supports MACsec link layer device-to-device security by using Cisco TrustSec Network Device Admission Control (NDAC) and the Security Association Protocol (SAP) key exchange. Link layer security includes both packet authentication between devices and MACsec encryption between devices (encryption is optional).

MACsec Standard Encryption

The Media Access Control Security (MACsec) standard provides data link layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) protocol provides the required session keys and manages the encryption keys. MKA and MACsec are implemented after a successful authentication by using the 802.1X Extensible Authentication Protocol (EAP) framework. Only host-facing links (links between network access devices and endpoint devices such as a PC or an IP phone) can be secured using MACsec.

A device that uses MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the client. MACsec frames are encrypted and protected with an integrity check value (ICV). When the device receives frames from the client, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The device compares the calculated value of the ICV to the ICV within the frame. If they are not identical, the frame is dropped. The device also encrypts and adds an ICV to any frame that is sent over a secured port (the access point used to provide the secure MAC service to a client) using the current session key.

The MKA protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1X-2010. The MKA protocol extends 802.1X to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by peers.

EAP Implementation of MKA

The Extensible Authentication Protocol (EAP) framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) that is shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). Because the device is the authenticator, it is also the key server, generating a random 128-bit secure association key (SAK), which it sends it to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the device sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes and no MKPDU is received from a participant. For example, if a client disconnects, the participant on the device continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the client.

eEdge Integration with MACsec

The Media Access Control Security (MACsec) standard is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The eEdge Integration with MACsec feature allows you to integrate the MACsec standard with enterprise edge (eEdge) devices to enhance Session Aware Networking capabilities. Session Aware Networking provides a policy and identity-based framework for edge devices to deliver flexible and scalable services to subscribers.

How to Configure eEdge Integration with MACsec

Integrating eEdge with MACsec

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	service-template <i>template-name</i> Example: Device(config)# service-template dot1x-macsec-policy	Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode.
Step 4	linksec policy {must-not-secure must-secure should-secure} Example: Device(config-service-template)# linksec policy must-secure	Sets the link security policy as must-secure. • Must-secure policy authorizes the eEdge device port only if a secure MACsec session is established.
Step 5	exit Example: Device(config-service-template)# exit	Exits service template configuration mode and returns to global configuration mode.
Step 6	policy-map type control subscriber <i>control-policy-name</i> Example: Device(config)# policy-map type control subscriber cisco-subscriber	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 7	event authentication-success [match-all match-any] Example: Device(config-event-control-policymap)# event authentication-success match-all	Specifies the type of event that triggers actions in a control policy if all authentication events are a match and enters control policy-map class configuration mode.
Step 8	priority-number class {control-class-name always} [do-all do-until-failure do-until-success] Example:	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.

	Command or Action	Purpose
	Device (config-class-control-policymap) # 10 class always do-until-failure	
Step 9	action-number activate {policy type control subscriber control-policy-name service-template template-name [aaa-list list-name] [precedence [replace-all]]} Example: Device (config-action-control-policymap) # 10 activate service-template dot1x-macsec-policy	Activates a control policy on a subscriber session.
Step 10	end Example: Device (config-action-control-policymap) # end	Exits control policy-map action configuration mode and enters privileged EXEC mode.

Identifying Link Layer Security Failures

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	class-map type control subscriber {match-all match-any match-none} control-class-name Example: Device (config) # class-map type control subscriber match-all linksec-failed	Creates a control class, which defines the conditions under which the actions of a control policy are executed and enters control class-map filter configuration mode.
Step 3	match authorization-failure {domain-change-failed linksec-failed} Example: Device (config-filter-control-classmap) # match authorization-failure linksec-failed	Configures a match condition in a control class based on the type of authorization failure received from an authorization failed event of a link layer security failure.
Step 4	exit Example: Device (config-class-control-policymap) # exit	Exits control class-map filter configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	policy-map type control subscriber <i>control-policy-name</i> Example: <pre>Device(config)# policy-map type control subscriber cisco-subscriber</pre>	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 6	event authentication-failure [match-all match-any] Example: <pre>Device(config-event-control-policymap)# event authentication-failure match-all</pre>	Specifies the type of event that triggers actions in a control policy if session authentication fails and enters control policy-map class configuration mode.
Step 7	priority-number class {control-class-name always } [do-all do-until-failure do-until-success] Example: <pre>Device(config-class-control-policymap)# 10 class linksec-failed do-until-failure</pre>	Specifies that the control class must execute the actions in a control policy, in the specified order, until one of the actions fails and enters control policy-map action configuration mode.
Step 8	end Example: <pre>Device(config-action-control-policymap)# end</pre>	Exits control policy-map action configuration mode and enters privileged EXEC mode.

Configuration Examples for eEdge Integration with MACsec

Example: Integrating eEdge with MACsec

```
Device> enable
Device# configure terminal
Device(config)# service-template dot1x-macsec-policy
Device(config-service-template)# linksec policy must-secure
Device(config-service-template)# exit
Device(config)# policy-map type control subscriber cisco-subscriber
Device(config-event-control-policymap)# event authentication-success match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 activate service-template dot1x-macsec-policy
Device(config-action-control-policymap)# end
```

Example: Identifying Linksec Failures

```
Device# configure terminal
Device(config)# class-map type control subscriber match-all linksec-failure
Device(config-filter-control-classmap)# match authorization-failure linksec-failed
Device(config-class-control-classmap)# exit
Device(config)# policy-map type control subscriber cisco-subscriber
Device(config-event-control-policymap)# event authentication-failure match-all
```

Example: Identifying Linksec Failures

```
Device(config-class-control-policymap)# 10 class linksec-failed do-until-failure
Device(config-action-control-policymap)# end
```