



WAN MACSEC and MKA Support Enhancements

The WAN MACsec and MKA feature introduces MACsec support on WAN, and uplink support and Pre-shared key support for the Macsec Key Agreement protocol (MKA).

- [Feature Information for WAN MACsec and MKA, on page 1](#)
- [Finding Feature Information, on page 2](#)
- [Prerequisites for WAN MACsec and MKA Support Enhancements, on page 2](#)
- [Restrictions for WAN MACsec and MKA Support Enhancements, on page 2](#)
- [Information About WAN MACsec and MKA Support Enhancements, on page 3](#)
- [How to Configure WAN MACsec and MKA Support Enhancements, on page 10](#)
- [Configuration Examples for WAN MACsec and MKA, on page 19](#)
- [Additional References, on page 25](#)

Feature Information for WAN MACsec and MKA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for WAN MACsec and MKA

| Feature Name | Releases | Feature Information |
|-------------------------------|----------------------------|--|
| WAN MACsec and MKA | Cisco IOS XE Release 3.14S | The WAN MACsec and MKA feature introduces MACsec support on WAN and uplink support and pre-shared key support for the MACsec Key Agreement protocol (MKA). The following commands were introduced or modified: confidentiality-offset, eapol destination-mac, key-server, linksec policy, replay-protection window-size . |
| MACsec on WAN Interface Cards | Cisco IOS XE Release 3.16S | The MACsec on WAN Interface Cards feature introduces MACsec support on WAN interface cards on Cisco 4000 Series Integrated Services Routers (ISRs). |

| Feature Name | Releases | Feature Information |
|---|----------------------------|---|
| MACsec CLI Option to Change EAPoL Frame Ethernet Type | Cisco IOS XE Release 3.17S | <p>The MACsec CLI Option to Change EAPoL Frame Ethernet Type feature provides a configuration option to allow users to change the Extensible Authentication Protocol over LAN (EAPoL) Frame Ethernet Type.</p> <p>The following commands were introduced or modified: eapol eth-type.</p> |

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for WAN MACsec and MKA Support Enhancements

- WAN MACsec requires MACsec license. See Table 8 in the document titled *Cisco ASR 1000 Series Ethernet Line Cards Data Sheet* – <https://www.cisco.com/c/en/us/products/collateral/application-networking-services/wide-area-application-services-waas-software/data-sheet-c78-729778.html>
- The Cisco ISR 4000 platforms require HSECK9 license to configure MACsec.
- Layer 2 transparent Ethernet Services must be present.
- The service provider network must provide a MACsec Layer 2 Control Protocol transparency such as, Extensible Authentication Protocol over LAN (EAPoL).

Restrictions for WAN MACsec and MKA Support Enhancements

- On Cisco ASR 1000 Series Aggregation Services Routers, MACsec does not support AAA accounting.
- MACsec is supported up to line rate on each interface. However, the forwarding capability may be limited by the maximum system forwarding capability.
- On the Cisco ASR1001-X router, MACsec is supported on the built-in ports only. It cannot be enabled on a Shared Port Adapter (SPA) that is installed on the router.
- MACsec configuration on Ether Channel (Link bundling) is not supported.
- Any interface configured with MACsec cannot be part of Ether Channel.

- MACsec configured on the native subinterface with the command **macsec dot1q-in-clear 1** on the main interface is not supported.
- From Cisco IOS XE Denali 16.3.3 release onwards, during RP Switchover, re-entry of macsec commands in physical/sub-interface configuration mode is not required.
- If the MKA session is torn down because of key unwrap failure, re-configure the pre-shared key based MKA session using MACsec configuration commands on the respective interfaces to bring the MKA session up.
- MACsec-configured on physical interface with Ethernet Virtual Circuits (EVC) is not supported. The EAPoL frames will get dropped in such cases.
- On Cisco ASR 1000 Series Aggregation Services Routers, the following table lists the GigabitEthernet interface and the maximum number of peers that are supported per interface:

| GigabitEthernet Interface | Peers per Interface |
|---------------------------|---------------------|
| 1G | 8 |
| 10G | 32 |
| 40G | 60 |
| 100G | 120 |

- When `macsec dot1q-in-clear` is enabled, the native VLAN is not supported.

Information About WAN MACsec and MKA Support Enhancements

MACsec and MKA Overview

MACsec is an IEEE 802.1AE standards based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec, 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. Only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

The 802.1AE encryption with MACsec Key Agreement (MKA) is supported on downlink ports for encryption between the routers or switches and host devices.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet.

To provide MACsec services over the WAN or Metro Ethernet, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

The packet body in an EAP-over-LAN (EAPoL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participant after 3 heartbeats (each heartbeat is of 2 seconds), peers are deleted from the live peer list. For example, if a client disconnects, the participant on the switch continues to operate MKA until 3 heartbeats have elapsed after the last MKPDU is received from the client.

The MKA feature support provides tunneling information such as VLAN tag (802.1Q tag) in the clear so that the service provider can provide service multiplexing such that multiple point-to-point or multipoint services can co-exist on a single physical interface and differentiated based on the now visible VLAN ID.

In addition to service multiplexing, VLAN tag in the clear also enables service providers to provide quality of service (QoS) to the encrypted Ethernet packet across the SP network based on the 802.1P (CoS) field that is now visible as part of the 802.1Q tag.

Benefits of WAN MACsec and MKA Support Enhancements

- Support for Point-to-point (P2P) deployment models.
- Support for Point-to-Multipoint (P2MP) deployment models.
- Support for multiple P2P and P2MP deployments on the same physical interface.
- Support for 128- and 256-bit Advanced Encryption Standard–Galois Counter Mode (AES-GCM) encryption for data packets.
- Support for 128- and 256-bit Advanced Encryption Standard-Cipher-based Message Authentication Code (AEC-CMAC) encryption for control packets.
- Support for VLAN tag in the clear option to enable Carrier Ethernet Service Multiplexing.
- Support for coexisting of MACsec and Non-MACsec subinterfaces.
- Support for configurable Extensible Authentication Protocol over LAN (EAPoL) destination address.
- Support for configurable option to change the EAPoL Ethernet type.
- Support for configurable replay protection window size to accommodate packet reordering in the service provider network.
- Support for MacSec stateless switchover. During route processor (RP) switchover on dual RP setup, there is a teardown of existing MacSec session and the session is re-negotiated/reinitiated automatically (stateless switchover). During this process, some traffic drop might occur for a few seconds. MacSec stateless switchover is supported from Cisco IOS XE Everest 16.6 Release onwards.

Best Practices for Implementing WAN MACsec and MKA Support Enhancements

- Ensure basic Layer 2 Ethernet connectivity is established and verified before attempting to enable MACsec. Basic ping between the customer edge devices must work.
- When you are configuring WAN MACsec for the first time, ensure that you have out of band connectivity to the remote site to avoid locking yourself out after enabling MACsec, if the session fails to establish.

- We recommend that you configure the **access-control should-secure** command while enabling MACsec for the first time and subsequently remove the command to change to default **access-control must-secure**, once the session establishment is successful, unless it is needed for migration.
- We recommend that you configure an interface MTU, adjusting it for MACsec overhead, for example, 32 bytes. Although MACsec encryption and decryption occurs at the physical level and MTU is size does not effect the source or destination router, it may effect the intermediate service provider router. Configuring an MTU value at the interface allows for MTU negotiation that includes MACsec overhead.

MKA Policy Inheritance

On WAN routers, MKA policy is inherited and also it has a default value. When a new session is started, the following rules apply:

- If an MKA policy is configured on a subinterface, it will be applied when an MKA session is started.
- If an MKA policy is not configured on a subinterface, a policy that is configured on the physical interface is be applied at session start.
- If a MKA policy is not configured on a subinterface or physical interface, default policy is applied at session start.

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.

Use the **key chain** *name* **macsec** to configure the MACsec key chain.

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.



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

Encryption Algorithms for Protocol Packets

Cryptographic Algorithm selection for MKA control protocol packets encryption is as follows:

- Cryptographic Algorithm to encrypt MKA control protocol packets is configured as part of the key chain. There can be only one cryptographic algorithm configured per key chain.
- A key server uses the configured MKA cryptographic algorithm from the key chain that is used.
- All nonkey servers must use the same cryptographic algorithm as the key server.

If an MKA cryptographic algorithm is not configured, a default cryptographic algorithm of AES-CMAC-128 (Cipher-based Message Authentication Code with 128-bit Advanced Encryption Standard) is used.

Encryption algorithm for Data packets:

```
mka policy p1
macsec-cipher-suite [gcm-aes-128 | gcm-aes-256
```

Encryption algorithm for MKA Control packets

```
key chain <name> macsec
key 01
key-string <Hex string>
cryptographic-algorithm [aes-256-cmac | aes-128-cmac]
```

It is recommended to change data packets cipher suite in the key server for the cipher suite rollover to be seamless, if the nonkey servers have the same cipher-suite configured in the list or is with default configuration.

Access Control Option for Smoother Migration

When MACsec is enabled on an interface, the entire interface traffic is secured, by default. MACsec does not allow any unencrypted packets to be transmitted or received from the same physical interface. However, to enable MACsec on selected subinterfaces, an additional Cisco proprietary extension has been implemented to allow unencrypted packets to be transmitted or received from the same physical interface.

Use the **macsec access-control** {**must-secure** | **should-secure**} command to control the behavior of unencrypted packets.

- The **should-secure** keyword allows unencrypted packets from the physical interface or subinterfaces to be transmitted or received.
- The **must-secure** keyword does not allow unencrypted packets from physical interface or subinterfaces to be transmitted or received. All such packets are dropped except for MKA control protocol packets
- If MACsec is enabled only on selected subinterfaces, configure the **should-secure** keyword option on the corresponding interface.

The default configuration for MACsec on subinterfaces is **macsec access-control must-secure**. This option is enabled by default when the **macsec** command is configured on an interface.



Note The **macsec access-control should-secure** command can be configured only at the interface level and not the subinterface. Configuring this command allows unencrypted traffic on a secured MACsec session.



Note For non-MACsec subinterface, you must configure the **should-secure** option for traffic to pass.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MKA (MACsec Key Agreement) is used as the control protocol. MKA selects the cipher suite to be used for encryption and to exchange the required keys and parameters between peers.

MKA uses Extensible Authentication Protocol over LAN (EAPoL) as the transport protocol to transmit MKA messages. By default, EAPoL uses a destination multicast MAC address of 01:80:c2:00:00:03 to multicast packets to multiple destinations. EAPoL is a standards-based protocol and other authentication mechanisms such as IEEE 802.1X also use the same protocol. Devices in the service provider cloud might consume this packet (based on the destination multicast MAC address), and try to process the EAPoL packet and eventually drop the packet. This causes MKA session to fail.

Use the **eapol destination-address** command to change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider. This ensures that the service provider tunnels the packet like any other data packet instead of consuming them.



Note The EAPoL destination address can be configured independently on either physical or subinterface level. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on the subinterface overrides the inherited value or policy for that subinterface.

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 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 set to 64. Use the **macsec replay-protection window-size** command to change the replay window size. The range for window size is 0 to 4294967295.

The replay protection window may be set to zero to enforce strict reception ordering and replay protection.



Note A replay protection window can be configured independently on either physical interface or subinterface. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on subinterface overrides the inherited value or policy for that sub-interface.

MACsec on WAN Interface Cards

In Cisco IOS XE Release 3.16S, MACsec is introduced on WAN interface cards (NIM-2GE-CU-SFP and NIM-2GE-CU-SFP) on Cisco 4000 Series Integrated Services Routers (ISRs).

This WAN interface card is a two one-Gigabit Ethernet-port Next Generation WAN Interface Card.

The following platforms support the Next Generation WAN Interface Card:

- Cisco ISR 4451
- Cisco ISR4431
- Cisco ISR4351
- Cisco ISR 4331

- Cisco ISR 4321

OIR Support

When a WAN interface card is operationally inserted or removed (OIR), the configuration associated with that interface is preserved such that if the interface is ever reinserted into the system it appears with the same configuration. However, in Cisco IOS XE Release 3.16s on Cisco ISR routers the following limitations apply for MACsec and MKA sessions:

- In some scale scenarios, after OIR MKA/MACsec session may be lost.
- MKA/MACsec session must be reestablished after OIR.

MACsec Performance on Cisco 4000 Series Integrated Services Routers

Table 2: Performance Numbers on Cisco ISR 4451 Router

| Frame Size | NDR per Port (pps) | Line Rate (%) | Module CPU (%) | Host CPU (%) |
|------------|--------------------|---------------|----------------|--------------|
| 64 | 1,077,532 | 72.41 | 44 | 65 |
| 128 | 692,568 | 82 | 29 | 42 |
| 256 | 405,797 | 89.6 | 17 | 25 |
| iMIX | 296,500 | 90.57 | 13 | 24 |
| 512 | 221,615 | 94.32 | 9 | 14 |
| 1024 | 116,163 | 97.02 | 5 | 7 |
| 1518 | 79,609 | 97.95 | 3.5 | 5 |
| 9000 | 13,808 | 99.64% | 1 | 2 |

MACsec Performance on Cisco ASR 1000 Platforms

The following tables show the performance numbers on Cisco ASR 1000 routers from Cisco IOS XE 16.6 release onwards.

Table 3: Performance Numbers on Cisco ASR1001-X Router

| Frame Size | Aggregate Rate Bits (bps) | Line Rate per port (%) | ESP CPU (%) |
|------------|---------------------------|------------------------|-------------|
| 64 | 10064767891.17 | 65.59 | 93.33 |
| iMIX | 17763891467.40 | 93.14 | 26 |
| 1418 | 19311044388.60 | 97.89 | 9 |

Table 4: Performance Numbers on Cisco ASR1001-HX Router

| Frame Size | Aggregate Rate Bits (bps) | Line Rate per port (%) | ESP CPU (%) |
|------------|---------------------------|------------------------|-------------|
| 64 | 28681245486.53 | 65.59 | 99 |
| iMIX | 65019905182.40 | 93.14 | 42 |
| 1418 | 64975057119.60 | 97.89 | 11 |

Table 5: Performance Numbers on Cisco ASR1002-HX Router

| Frame Size | Aggregate Rate Bits (bps) | Line Rate per port (%) | ESP CPU (%) |
|------------|---------------------------|------------------------|-------------|
| 64 | 51467063849.50 | 65.59 | 96 |
| iMIX | 105267526427 | 93.14 | 36 |
| 1418 | 100007152449 | 97.89 | 10 |

MACsec Compatibility Matrix for ASR 1000 and ISR 4400 Platforms

| Platform | Built-In Ports | EPA-18x1GE | EPA-10x10GE | EPA-1x40GE / EPA-2x40GE | NIM-2GE-CU-SFP |
|------------|-------------------------------------|-------------------------------------|---|----------------------------------|------------------------------|
| ASR1001-X | Cisco IOS XE Release 3.13.1S | NA | NA | NA | NA |
| ASR1001-HX | Cisco IOS XE Everest Release 16.4.1 | NA | NA | NA | NA |
| ASR1002-HX | Cisco IOS XE Denali Release 16.3.1 | Cisco IOS XE Denali Release 16.3.1 | Cisco IOS XE Denali Release 16.3.2 / 16.4.1 | Cisco IOS XE Fuji Release 16.8.1 | NA |
| ASR1006-X | NA | Cisco IOS XE Everest Release 16.4.1 | Cisco IOS XE Denali Release 16.3.1 | Cisco IOS XE Fuji Release 16.8.1 | NA |
| ASR1009-X | NA | Cisco IOS XE Everest Release 16.4.1 | Cisco IOS XE Denali Release 16.3.1 | Cisco IOS XE Fuji Release 16.8.1 | NA |
| ASR1013 | NA | Cisco IOS XE Everest Release 16.4.1 | Cisco IOS XE Denali Release 16.3.1 | Cisco IOS XE Fuji Release 16.8.1 | NA |
| ISR44XX | NA | NA | NA | NA | Cisco IOS XE Release 3.16.0S |
| ISR43XX | NA | NA | NA | NA | Cisco IOS XE Release 3.16.0S |

| Platform | Built-In Ports | EPA-18x1GE | EPA-10x10GE | EPA-1x40GE / EPA-2x40GE | NIM-2GE-CU-SFP |
|----------|----------------------------------|------------|-------------|-------------------------|------------------------------|
| ISR4462 | Cisco IOS XE Fuji Release 16.9.1 | NA | NA | NA | Cisco IOS XE Release 3.16.0S |

**Note**

- GLC-100FX is not supported.
- MIP-100 is required for ASR1006X, ASR1009X, and ASR1013 platforms for EPA18x1GE, EPA-10x10GE, EPA-1x40GE, and EPA-2x40GE.
- MACsec on ASR1001-X requires IPsec license.
- MACsec on ASR1001-HX, ASR1002-HX, and EPAs require per port MACsec licenses.
- The Cisco ISR 4000 platforms require HSECK9 license to configure MACsec.

How to Configure WAN MACsec and MKA Support Enhancements

Configuring MKA

The MACsec Key Agreement (MKA) enables configuration and control of keying parameters. Perform the following task to configure MKA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy *policy-name***
4. **include-icv-indicator**
5. **key-server priority *key-server-priority***
6. **macsec-cipher-suite {gcm-aes-128 | gcm-aes-256 | gcm-aes-xpn-128 | gcm-aes-xpn-256}**
7. **sak-rekey interval *interval***
8. **confidentiality-offset 30**
9. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|----------------------------------|--|
| Step 1 | enable Example: | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |

| | Command or Action | Purpose |
|---------------|---|--|
| | Device> enable | |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | mka policy <i>policy-name</i> Example: Device(config)# mka policy MKAPolicy | Configures an MKA policy. |
| Step 4 | include-icv-indicator Example: Device(config-mka-policy)# include-icv-indicator | (Optional) Include ICV indicator in MKPDU. |
| Step 5 | key-server priority <i>key-server-priority</i> Example: Device(config-mka-policy)# key-server priority 200 | (Optional) Configures MKA key server priority. |
| Step 6 | macsec-cipher-suite {gcm-aes-128 gcm-aes-256 gcm-aes-xpn-128 gcm-aes-xpn-256} Example: Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128 gcm-aes-256 | (Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order. |
| Step 7 | sak-rekey interval <i>interval</i> Example: Device(config-mka-policy)# sak-rekey interval 30 | (Optional) Sets the SAK rekey interval (in seconds). The range is from 30 to 65535, and the default value is 0. The SAK rekey timer does not start by default until it is configured. <ul style="list-style-type: none"> To stop the SAK rekey timer, use the no sak-rekey interval command under the defined MKA policy. |
| Step 8 | confidentiality-offset 30 Example: Device(config-mka-policy)# confidentiality-offset 30 | (Optional) Configures confidentiality offset for MACsec operation. |
| Step 9 | end Example: Device(config-mka-policy)# end | Returns to privileged EXEC mode. Note The MKA policy does not process confidentiality offset for XPN ciphers. Therefore when both XPN and non-XPN ciphers are configured in an MKA policy alongwith confidentiality offset, the confidentiality offset is ignored for XPN ciphers. It is therefore strongly recommended to use your discretion while using configuring a MKA policy with XPN or non-XPN ciphers. |

Example

You can use the **show mka policy** command to verify the configuration. Here's a sample output of the **show** command. If you do not want to include icv-indicator in MKPDUs, use the **no include-icv-indicator** command in the MKA policy.

MKA Policy Summary...

Codes : CO - Confidentiality Offset, ICVIND - Include ICV-Indicator,
SAKR OLPL - SAK-Rekey On-Live-Peer-Loss,
DP - Delay Protect, KS Prio - Key Server Priority

| Policy Name | KS Prio | DP | CO | SAKR OLPL | ICVIND | Cipher Suite(s) | Interfaces Applied |
|------------------|---------|-------|----|-----------|--------|----------------------------|--------------------|
| *DEFAULT POLICY* | 0 | FALSE | 0 | FALSE | TRUE | GCM-AES-128 GCM-AES-256 | N/A |
| confid50 | 0 | FALSE | 50 | FALSE | TRUE | GCM-AES-128 GCM-AES-256 | |
| icv | 0 | FALSE | 0 | FALSE | TRUE | GCM-AES-128 GCM-AES-256 | Te3/0/9 |
| k10 | 0 | FALSE | 0 | FALSE | TRUE | GCM-AES-128 GCM-AES-256 | |
| xpn128 | 0 | FALSE | 0 | FALSE | TRUE | GCM-AES-XPB-128 | Fo2/1/1 |

Configuring MACsec and MKA on Interfaces

Perform the following task configure MACsec and MKA on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mka policy** *policy-name*
5. **mka pre-shared-keykey-chainkey-chain-name**
6. **macsec**
7. **macsec replay-protection window-size**
8. **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 | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--|--|
| | Example: Device# configure terminal | |
| Step 3 | interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 0/0/0 | Enters interface configuration mode. |
| Step 4 | mka policy <i>policy-name</i> Example: Device(config-if)# mka policy MKAPolicy | Configures an MKA policy |
| Step 5 | mka pre-shared-key key-chain <i>key-chain-name</i> Example: Device(config-if)# mka pre-shared-key key-chain key-chain-name | Configures an MKA pre-shared-key key-chain keychain1 Note The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces. |
| Step 6 | macsec Example: Device(config-if)# macsec | Configures MACsec for the EAPOL frame ethernet type. |
| Step 7 | macsec replay-protection window-size Example: Device(config-if)# macsec replay-protection window-size 10 | Sets the MACsec window size for replay protection. |
| Step 8 | end Example: Device(config-if)# end | Returns to privileged EXEC mode. |

Configuring MKA Pre-shared Key

Perform the following task to configure MACsec Key Agreement (MKA) pre-shared key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *key-chain-name* [**macsec**]
4. **key** *hex-string*
5. **cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**}
6. **key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
7. **lifetime local** {{*day month year duration seconds*}
8. **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 | key chain <i>key-chain-name</i> [macsec] Example: Device(config)# Key chain keychain1 macsec | Configures a key chain and enters keychain configuration mode |
| Step 4 | key <i>hex-string</i> Example: Device(config-keychain)# key 9ABCD | Configures a key and enters keychain key configuration mode. Note From Cisco IOS XE Everest Release 16.6.1 onwards, the Connectivity Association Key name (CKN) uses exactly the same string, which is configured as the hex-string for the key. For more information about this behavior change, see the section titled "MKA-PSK: CKN Behavior Change" after this task. |
| Step 5 | cryptographic-algorithm {gcm-aes-128 gcm-aes-256} Example: Device(config-keychain-key)# cryptographic-algorithm gcm-aes-128 | Set cryptographic authentication algorithm. |
| Step 6 | key-string {[0 6] <i>pwd-string</i> 7 <i>pwd-string</i>} Example: Device(config-keychain-key)# key-string 0 pwd | Sets the password for a key string. |
| Step 7 | lifetime local {{<i>day month year duration seconds</i>} Example: Device(config-keychain-key)# lifetime local 16:00:00 Nov 9 2014 duration 6000 | Sets the lifetime for a key string. The range you can specify for the duration is between 1 and 864000 seconds. |
| Step 8 | end Example: Device(config-keychain-key)# end | Returns to privileged EXEC mode. |

Example for Connectivity Association Key (CAK) Rekey

CAK rekey will happen in the following cases:

- When moving from Key 01 to Key 02 within the Key Chain K1.
- When moving from one Key Chain K1 to another Key Chain K2.

Note: It is recommended to configure keys such that there is an overlap between the lifetime of the keys so that CAK rekey is successful and there is a seamless transition between the Keys/CA (without any traffic loss or session restart)

```
Device# show key chain k1
Key-chain k1:
MacSEC key chain
key 01 - text "c890433ale05ef42d723a6b58af8fdbf7a25f42b3cda6a5eeb5ae4bf3a0a679f"
lifetime (00:00:00 UTC Oct 29 2014) - (12:10:00 UTC Oct 29 2014)
key 02 - text "14d9167d538819405c0ff78c655141ed4b3c7242562c0fb0f7a56f780bf29e52"
lifetime (12:00:00 UTC Oct 29 2014) - (18:05:00 UTC Oct 29 2014)
key 03 - text "88d971cb19d9f2598ad76edc562ade2e7e91e3ed70524f5c3c4d8d9599d0670e"
lifetime (18:00:00 UTC Oct 29 2014) - (18:10:00 UTC Oct 29 2014)
key 04 - text "75474bce819b49ad7e5bd06236bc0c944c69892f71e942e2f9812b7d3a7b2a5f"
lifetime (18:10:00 UTC Oct 29 2014) - (infinite)

!In this case, Key 01, 02, 03 have overlapping time, but not key 04. Here is the sequence,
how this works:
@00:00:00 - A new MKA session is Secured with key 01
@12:00:00 - CAK Rekey triggers with key 02 and upon success goes to Secured state
@18:00:00 - CAK Rekey triggers with key 03 and upon success goes to Secured state
@18:10:00 - Key 03 dies, hence MKA session using this key is brought down
@18:10:00 - Key 04 becomes active and a new MKA session is triggered with this key. Upon
success, session will be Secured and UP for infinite time.
```

MKA-PSK: CKN Behavior Change

From Cisco IOS XE Everest Release 16.6.1 onwards, for MKA-PSK sessions, instead of fixed 32 bytes, the Connectivity Association Key name (CKN) uses exactly the same string as the CKN, which is configured as the hex-string for the key.

Example Configuration:

```
configure terminal
key chain abc macsec
key 11
cryptographic-algorithm aes-128-cmac
key-string 12345678901234567890123456789013
lifetime local 12:21:00 Sep 9 2015 infinite

end
```

For the above example, the following will be the **show** command output for the **show mka session** command:

```
Device# show mka session

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


| | Command or Action | Purpose |
|---------------|--|---|
| | Device> enable | |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface type number Example: Device(config)# interface GigabitEthernet 0/0/1 | Enters interface configuration mode. |
| Step 4 | eapol eth-type Example: Device(config-if)# eapol eth-type 0xB860 | Configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface. Note From Cisco IOS Release XE 3.17, the macsec eth-type command has been replaced by the eapol eth-type command. |
| Step 5 | exit Example: Device(config-if)# exit | Exits interface configuration mode and returns to global configuration mode. |

Configuring Destination MAC Address on Interface and Sub-interface

Perform the following task to configure destination MAC address on the Interface or Subinterface. The destination MAC could be the MAC of the peer or a multicast MAC address. When the **eapol destination-address** command is configured on the main interface, it is applied to any subinterfaces on that interface. However, if the **eapol destination-address** command is configured on the subinterface, that takes take precedence over the command on the main interface.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface type number
4. eapol destination-address [MAC-Address | [bridge-group-address | broadcast-address | lldp-multicast-address]
5. eapol destination-address bridge-group-address
6. eapol destination-address broadcast-address
7. eapol destination-address lldp-multicast-address
8. end

DETAILED STEPS

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

Configuring Destination MAC Address on Interface and Sub-interface

| | Command or Action | Purpose |
|---------------|---|---|
| | Example: Device> enable | <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface type number Example: Device(config)# interface GigabitEthernet 0/0/1 | Enters interface configuration mode. |
| Step 4 | eapol destination-address [MAC-Address [bridge-group-address broadcast-address lldp-multicast-address] Example: Device(config-if)# eapol destination-address 0018.b967.3cd0 | Configures an Extensible Authentication Protocol over LAN (EAPoL) destination MAC address on the interface. |
| Step 5 | eapol destination-address bridge-group-address Example: Device(config-if)# eapol destination-address bridge-group-address | Sets the destination address as a bridge group. |
| Step 6 | eapol destination-address broadcast-address Example: Device(config-if)# eapol destination-address broadcast-address | Sets the destination address as a broadcast address. |
| Step 7 | eapol destination-address lldp-multicast-address Example: Device(config-if)# eapol destination-address lldp-multicast-address | Sets the destination address as a LLDP multicast address. |
| Step 8 | end Example: DeviceDevice(config-if)# end | Returns to privileged EXEC mode. |

Configuration Examples for WAN MACsec and MKA

Example: Point-to-point, CE to CE Connectivity Using EPL Service

The following is the sample configuration for point-to-point, Customer Edge to Customer Edge connectivity using Ethernet Private Line (EPL) using port-based service.

```
!Customer Edge 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!Customer Edge 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
```

Example: Point-to-point, Hub and Spoke Connectivity using EVPL Service

The following is sample configuration for point-to-point, hub and spoke connectivity using Ethernet Virtual Private Line (EVPL) Service in VLAN mode.

```
!CE 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 10
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
interface GigabitEthernet0/0/4.2
  encapsulation dot1Q 20
  ip address 10.3.2.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
  macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
```

```

encapsulation dot1Q 10
 ip address 10.3.1.1 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*

!CE 3
key chain k1 macsec*
 key 01
 key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec replay-protection-window-size 100
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.1 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*

```



Note All commands with asterix (*) are mandatory commands.

Example: Point-to-point, Hub and Spoke Connectivity with MACsec and non-MACsec Spokes

The following is sample output of point-to-point, Hub and Spoke Connectivity with MACsec and non-MACsec spokes.

```

!CE1
key chain k1 macsec*
 key 01
 key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec access-control should-secure*
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.1 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*
interface GigabitEthernet0/0/4.2
 encapsulation dot1Q 20
 ip address 10.3.2.1 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*
interface GigabitEthernet0/0/4.3
 encapsulation dot1Q 30
 ip address 10.3.3.1 255.255.255.0

!CE2
key chain k1 macsec*
 key 01
 key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec access-control should-secure*
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.2 255.255.255.0

```

```

mka pre-shared-key key-chain k1*
macsec*

!CE3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  macsec dot1q-in-clear 1*
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 20
  ip address 10.3.2.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!CE4
interface GigabitEthernet0/0/4.1
  encapsulation dot1Q 30
  ip address 10.3.3.2 255.255.255.0

```

Example: Multipoint-to-multipoint, Hub and Spoke connectivity using EP-LAN Service

The following example shows sample configuration multipoint-to-multipoint, hub and Spoke connectivity using Ethernet Private LAN (EP-LAN) Service in port mode.

```

!CE 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/4
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  mka policy p1
  macsec*

!CE 2
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/4
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  mka policy p1
  macsec*

!CE 3
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac
mka policy p1
  macsec-cipher-suite gcm-aes-256

```

```
interface GigabitEthernet0/0/4
 ip address 10.3.1.3 255.255.255.0
 mka pre-shared-key key-chain k1*
 mka policy p1
 macsec*
```

Example: Multipoint-to-multipoint, Hub and Spoke Connectivity Using EVP-LAN Service

The following is sample configuration for multipoint-to-multipoint, hub and spoke connectivity using Ethernet Virtual Private LAN (EVP-LAN) Service in VLAN mode:

```
!CE 1
key chain k1 macsec*
 key 01
 key-string 123456789012345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec replay-protection-window-size 100
 eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.1 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*

!CE 2
key chain k1 macsec*
 key 01
 key-string 123456789012345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec replay-protection-window-size 100
 eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.2 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*

!CE 3
key chain k1 macsec*
 key 01
 key-string 123456789012345678901234567890123456789012
interface GigabitEthernet0/0/4
 macsec dot1q-in-clear 1*
 macsec replay-protection-window-size 100
 eapol destination-address broadcast
interface GigabitEthernet0/0/4.1
 encapsulation dot1Q 10
 ip address 10.3.1.3 255.255.255.0
 mka pre-shared-key key-chain k1*
 macsec*
```

Example: Performing Maintenance Tasks Without Impacting Traffic

The following are sample configurations of performance maintenance tasks that do not impact traffic:

Changing a Pre-Shared Key (CAK Rollover)

The following is sample configuration for changing a pre-shared key:



Note Keys can be configured to automatically roll over to the next key by configuring a lifetime on both routers.

```
!From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012

!To
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
  lifetime local 10:30:00 Oct 30 2014 11:30:00 Oct 30 2014
  key 02
  key-string 11145678901234567890123456789012
```

Changing a Key Chain (Keychain Rollover)

The following is the sample configuration for changing a key chain—Keychain Rollover

```
! From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface TenGigabitEthernet0/0/0.10
  mka pre-shared-key key-chain k1

! To
key chain k1 macsec
  key 01
  key-string 12345678901234567890123456789012
key chain k2 macsec
  key 02
  key-string abcdef0987654321abcdef0987654321
interface TenGigabitEthernet0/0/0.10
  mka pre-shared-key key-chain k2
```



Note The defined key ID, under any key chain, should be a unique value on the device.

A router can become a key server by configuring a lower priority than other peer routers that participate in the same session. Configure a key server priority so that the key server selection is deterministic. For example, in a Hub and Spoke scenario, the most ideal place for a key server is the Hub site router.

```
!Hub Site (Key Server):
mka policy p1
key-server priority 0
!0 is the default.

interface TenGigabitEthernet0/0/0.10
  mka pre-shared-key key-chain k1
  mka policy p1
```

Example: Performing Maintenance Tasks Without Impacting Traffic

```
!Spoke Sites (non-Key Servers):
mka policy p1
key-server priority 1

interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k1
mka policy p1
```

The following is sample configuration for changing Cipher Suite to encrypt data traffic:

```
mka policy p1
 macsec-cipher-suite gcm-aes-128
interface GigabitEthernet0/0/1.10
 mka policy p1

!Alternate configuration

mka policy p1
 macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/1.10
 mka policy p1

key chain k3 macsec
 key 01
  key-string abcdef0987654321abcdef0987654321
  cryptographic-algorithm aes-128-cmac
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k3

!Alternate configuration:

key chain k3 macsec
 key 01
  key-string abcdef0987654321abcdef0987654321
  cryptographic-algorithm aes-256-cmac
interface TenGigabitEthernet0/0/0.10
 mka pre-shared-key key-chain k3
```

EAPOL Destination MAC address can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the subinterfaces, if configured at the physical interface level. To override the inherited value, configure the MAC address at the subinterface mode. Default EAPOL destination MAC address is 01:80:c2:00:00:03.

```
interface TenGigabitEthernet0/0/0
 eapol destination-address <H.H.H>

!Alternate configuration

interface TenGigabitEthernet0/0/0
 bridge-group-address

!Alternate configuration

interface TenGigabitEthernet0/0/0
 lldp-multicast-address>

mka policy p1
 confidentiality-offset 30
interface GigabitEthernet0/0/1.10
 mka policy p1
```


Example: Performing Maintenance Tasks—Traffic Impacting

Changing a Replay Protection Window Size

Replay protection window can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the sub interfaces if configured at the physical interface level. If you need to override the inherited value, configure it at the subinterface mode. The default replay protection window size is 64.

```
interface TenGigabitEthernet0/0/0
macsec replay-protection window-size 10
```

```
interface TenGigabitEthernet0/0/0.10
macsec replay-protection window-size 5
```

Enabling or Disabling VLAN (dot1q) Tag in the Clear Option

The **macsec dot1q-in-clear** command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

```
interface GigabitEthernet0/0/1
macsec dot1q-in-clear 1
```

The **macsec access-control [must-secure | should-secure]** command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

```
interface GigabitEthernet0/0/1
macsec access-control must-secure/should-secure
```

Additional References

Related Documents

| Related Topic | Document Title |
|--------------------|--|
| Cisco IOS commands | Cisco IOS Master Command List, All Releases |
| Security commands | <ul style="list-style-type: none"> • Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z |

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 and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |