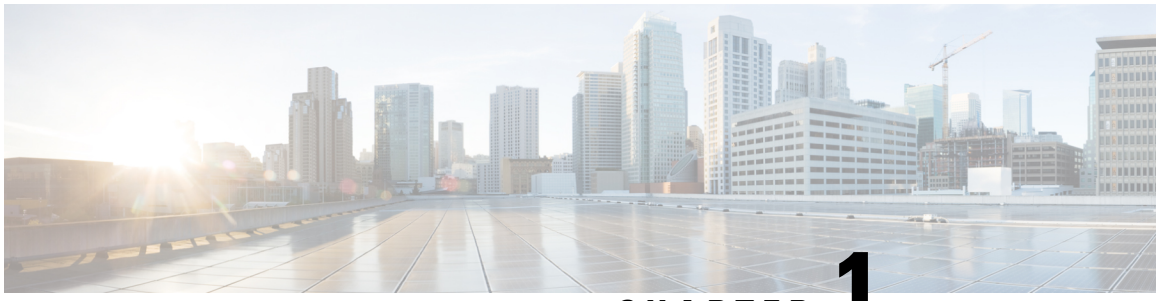
MACSEC and MKA Configuration Guide, Cisco IOS XE 17

First Published: 2014-12-17

Last Modified: 2023-02-17

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

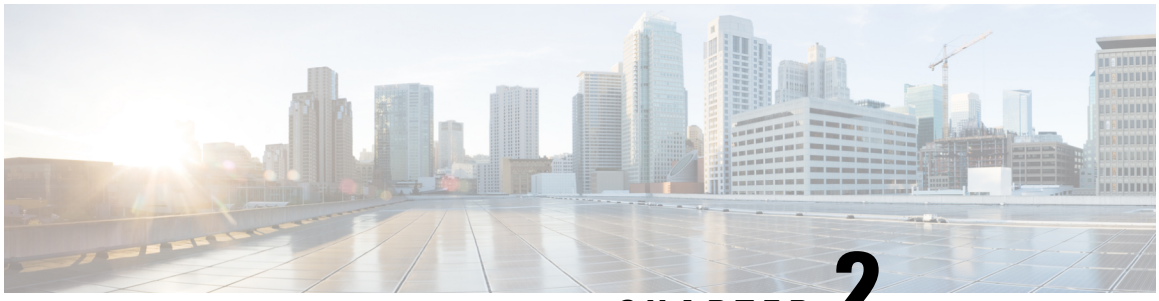
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.



CHAPTER 2

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

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 .

Feature Name	Releases	Feature Information
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).
MACsec CLI Option to Change EAPoL Frame Ethernet Type	Cisco IOS XE Release 3.17S	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. The following commands were introduced or modified: eapol eth-type.
Support for configuring port channel with MACsec encryption	Cisco IOS XE Gibraltar 17.2	This enhancement lets you configure port channels on MACsec-enabled interfaces for seamless flow of port-channel traffic. Thus, the traffic is secured.
MACSec Fallback Key Support	Cisco IOS XE Cupertino 17.8.1a	This feature introduces a fallback mechanism to re-establish the MACsec Key Agreement (MKA) session when it fails because of primary pre-shared key (PSK) mismatch. This fallback mechanism can be configured by using the **mka pre-shared-key key-chain** command.
MACsec Extended Packet Numbering (XPN)	Cisco IOS XE 16.8: 40-G and 100-G interfaces on Cisco 1000 Series Aggregation Services Routers Cisco IOS XE 17.3.2: 40-G and 100-G interfaces on Cisco Catalyst 8500 Series Edge Platforms Cisco IOS XE Cupertino 17.9.1: 10-G interfaces on Cisco Catalyst 8500 and 8500L Series Edge Platforms	This enhancement introduces support for extending the packet numbering for MACsec frames from 32-bit to 64-bit. To configure this enhancement, two cipher suites, gcm-aes-xpn-128 and gcm-aes-xpn-256, are introduced as keywords in the **macsec-cipher-suite** command.

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.
- From Cisco IOS XE Release 17.8.1, MKA supports the eth-type 888e or 876f for the EAPoL frame under the EFP interface of the physical interface.

Note You can only establish one MKA session for one EFP per interface when you configure eth-type 876f for the EAPoL frame under the EFP interface.

- From Cisco IOS XE Release 17.8.1, when you configure the **l2procol {forward|peer} dot1x** command on the Ethernet Virtual Circuits (EVC), it will be valid for the dot1x packet with both eth-type 888e and 876f.
- To configure port-channel, ensure that you configure MACsec at each interface of the link bundle.
- 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 participants after 3 hearbeats (each hearbeat 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.

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.

MACsec Fallback Preshared Key (PSK)

A MACsec key chain can have multiple preshared keys (PSKs) to establish MKA sessions. An MKA session can fail because of PSK mismatch. The fallback mechanism can re-establish the MKA session if the primary PSK fails to bring up the session.

A PSK-based MACsec session is established using the primary PSK defined in the MACsec key chain. This session can fail because of connectivity association key (CKN) mismatch. If a fallback key is configured, it establishes a fallback session whenever the primary PSK session does not get established. The fallback session prevents downtime, and allows you to fix the issue that is causing the primary PSK failure, without disrupting the session.

The process of fallback session is described here:

1. The primary PSKs configured on the participating devices do not match, and the Integrity Check Value (ICV) verification fails. ICV verification failure during the process of rekeying the Connectivity Association Key (CAK) leads to traffic drop.
2. If the fallback key chain is configured, the fallback PSK is activated.
3. The initial negotiation for the fallback key chain is completed. If the fallback key chains on participating devices match, a backup session is created and remains in standby mode.
4. When a primary key mismatch is encountered, this backup session is activated with a Secure Association Key (SAK) rekey.

Prerequisites for MACsec Fallback PSK

- The fallback key chains configured on peers must be identical.
- The lifetime of the fallback key configured for an interface must be valid.

- The keys in the primary key chain must have a minimum overlap time of 15 seconds.
- The CKNs in the primary and fallback key chains must be unique.

Restrictions for MACsec Fallback PSK

- There can be only one key in the fallback key chain; the lifetime for this key must not be configured and must remain in the default (infinite or unlimited) state. Multiple keys in the fallback key chain are not supported.
- The fallback key feature is not supported over Ethernet Virtual Circuit (EVC).
- The fallback key feature is not supported in the Point-to-Multi-Point (P2MP) topology.
- The fallback key feature is not supported in a high-availability (HA) setup.
- The MKA session is terminated with the lifetime expiry of the last key in a primary key chain.
- The primary key chain does not support keys with infinite lifetime. If an infinite-lifetime key is required, it must be configured as the last key in the PSK key chain.
- The MACsec Fallback PSK feature is only supported in Autonomous mode and not in Controller mode

Supported Platforms and Releases for MACsec Fallback PSK

From Cisco IOS XE Cupertino 17.8.1a, the MACsec Fallback PSK feature is supported on the following platforms:

- Cisco 1000 Series Aggregation Services Routers
- Cisco Catalyst 8500 Series Edge platforms

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.

MACsec eXtended Packet Numbering (XPN)

Every MACsec frame contains a 32-bit packet number. This packet number is specific for each Secure Association Key (SAK). SAK rekeying occurs when this packet number is exhausted (after sending 75 percent of 2^{31}- 1 packets). For high-capacity links associated with a SAK, the packet number gets exhausted within a few seconds, leading to frequent SAK rekeying. Frequent SAK rekeying can disrupt traffic and cause operational risks.

XPN support changes the packet number of the MACsec frame from 32-bit to 64-bit. This 64-bit XPN extends the packet number exhaustion timeframe for high-capacity links to 75 percent of (2^{63}- 1), thereby extending the validity to several years, and eliminates the need for frequent SAK rekeying in high-capacity links with 32-bit packet number.

To support this configuration, two cipher suites (gcm-aes-xpn-128 and gcm-aes-xpn-256) are introduced as keywords in the **macsec-cipher-suite** command.

Prerequisites for MACsec eXtended Packet Numbering (XPN)

There are no specific prerequisites for this feature.

Restrictions for MACsec eXtended Packet Numbering (XPN)

- The MACsec cipher announcement is not supported for MACsec XPN ciphers.
- MACsec XPN cipher suites do not provide confidentiality protection with a confidentiality offset, and can only be used for confidentiality protection without a confidentiality offset.

Supported Platforms and Releases for MACsec eXtended Packet Numbering (XPN)

This feature is supported on the following platforms:

Table 2: Supported Platforms and Releases for MACsec XPN

Platform	Supported Release
Cisco 1000 Series Aggregation Services Routers (40-G and 100-G interfaces)	Cisco IOS XE 16.8 onwards
Cisco Catalyst 8500 Series Edge Platforms (40-G and 100-G interfaces)	Cisco IOS XE 17.3.2 onwards
Cisco Catalyst 8500 and 8500L Series Edge Platforms (10-G interface)	Cisco IOS XE 17.9.1a onwards

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 3: 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 4: 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 5: 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 6: 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

Frame Size	Aggregate Rate Bits (bps)	Line Rate per port (%)	ESP CPU (%)
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
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.

Note Starting from IOS XE 17.2 Gibraltar, port-channel configuration is supported with MACsec. To configure this feature, ensure that you configure MACsec at each interface of the link bundle. For more information, see *Configuration Examples*

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. • 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** *policy-name* **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** *key-server-priority* **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** *interval* **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. • 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:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-mka-policy)# end`	**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            KS   DP    CO SAKR  ICVIND Cipher          Interfaces
Name              Prio          OLPL         Suite(s)        Applied
================================================================================
*DEFAULT POLICY*  0    FALSE 0  FALSE TRUE   GCM-AES-128     N/A
                                              GCM-AES-256

confid50          0    FALSE 50 FALSE TRUE   GCM-AES-128
                                              GCM-AES-256

icv               0    FALSE 0  FALSE TRUE   GCM-AES-128     Te3/0/9
                                              GCM-AES-256

k10               0    FALSE 0  FALSE TRUE   GCM-AES-128
                                              GCM-AES-256

xpn128            0    FALSE 0  FALSE TRUE   GCM-AES-XPN-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-chain***key-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** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Enters interface configuration mode.
Step 4	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy
Step 5	**mka pre-shared-keykey-chain***key-chain-name* **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 Preshared Key (PSK)

Perform the following task to configure MACsec Key Agreement (MKA) preshared 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** *key-chain-name* [**macsec**] **Example:** `Device(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode
Step 4	**key** *hex-string* **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**] *pwd-string*	**7**	*pwd-string*} **Example:** `Device(config-keychain-key)# key-string 0 pwd`	Sets the password for a key string.

	Command or Action	Purpose
Step 7	**lifetime local** {{*day month year* **duration** *seconds*} **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 "c890433a1e05ef42d723a6b58af8fdbf7a25f42b3cda6a5eeb5ae4bf3a0a679f"
        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 sesion 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.
```

Configuring MACsec Fallback PSK

To configure a MACsec fallback PSK, perform the following tasks:

Before you begin

Ensure that MACsec is enabled, and a primary and fallback keychain and key ID are configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface name*
4. **mka pre-shared-key key-chain** <*primary-psk-key-chain*> **fallback key-chain** <*fallback-key-chain*>
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**configure terminal**
	Example:
	`router# configure terminal`
	`router(config)#`
		Enters global configuration mode.

Step 3	**interface** *interface name*
	Example:
	`router(config)# interface Te 1/1/0`
	`router(config-if)#`
		Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot or port.

Step 4	**mka pre-shared-key key-chain** <*primary-psk-key-chain*> **fallback key-chain** <*fallback-key-chain*> **Example:** `router(config-if)# mka pre-shared-key key-chain ABC fallback-keychain fb_ABC`	Specifies the fallback keychain to use after a MACsec session failure becaisue of a key or key ID mismatch, or key expiration. The fallback key ID should not match any key ID from a primary keychain.
Step 5	**copy running-config startup-config** **Example:** `router(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration. In such a scenario, this step configures the primary and fallback PSKs to the startup configuration.

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

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

          Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server

          Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN


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

          Et0/0          aabb.cc00.6600/0002     icv            NO                NO

          2              aabb.cc00.6500/0002 1                  Secured           11    *Note
 that the CKN key-string is exactly the same that has been configured for the key as
hex-string.*
```

In the case of interoperability between two images -- one having the CKN behavior change, and one without the CKN behavior change, then the hex-string for the key must be a 64-character hex-string with zero padded to work on a device that has an image with the CKN behavior change. See the example below:

Configuration without CKN key-string behavior change:

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

Configuration with CKN key-string behavior change:

```
config t
  key chain abc macsec
   key 11000000000000000000000000000000000000000000000000000000000000000000000000000000
   cryptographic-algorithm aes-128-cmac
   key-string 12345678901234567890123456789013
   lifetime local 12:21:00 Sep 9 2015 infinite
```

Configuring MACsec eXtended Packet Numbering (XPN)

Perform the following tasks to configure XPN for MACsec:

Configure MKA Policy with XPN Cipher Suites

To configure XPN cipher suites in an MKA policy, perform the following tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **macsec-sipher-suite** {*gcm-aes-128* | *gcm-aes-256* | *gcm-aes-xpn-128* | *gcm-aes-xpn-256*}
5. **sak-rekey interval** *time-interval*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**configure terminal**
	Example:
	`router# configure terminal`
	`router(config)#`
		Enters global configuration mode.

Step 3	**mka policy** *policy-name* **Example:** `router(config)# mka policy Policy1`	Identifies an MKA policy, and enters MKA policy configuration mode. The maximum policy name length is 16 characters. **Note** The default MACsec cipher suite in an MKA policy is always **gcm-aes-128**. If the device supports both **gcm-aes-128** and **gcm-aes-256** ciphers, we recommend that you define and use a user-defined MKA policy to include both 128-bit and 256-bit ciphers or only 256-bit cipher, as may be required.
Step 4	**macsec-sipher-suite** {*gcm-aes-128* | *gcm-aes-256* | *gcm-aes-xpn-128* | *gcm-aes-xpn-256*} **Example:** `router(config-mka-policy)# macsec-cipher-suite gcm-aes-xpn-256`	Configures cipher suite for deriving SAK with 128-bit and 256-bit encryption for XPN.
Step 5	**sak-rekey interval** *time-interval* **Example:** `router(config-mka-policy)# sak-rekey interval 50`	(Optional) Configures the SAK rekey interval (in seconds). The range is from 30 to 65535. By default, the SAK rekey interval occurs automatically depending on the interface speed. Use the **no** form of this command to stop the SAK rekey timer.

	Command or Action	Purpose
Step 6	**end** **Example:** `router(config-mka-policy)# end`	Exits MKA policy configuration mode and returns to privileged EXEC mode.

Example

The following example shows the configuration of a MACsec XPN policy:

```
enable
configure terminal
mka policy mka-xpn-policy
macsec-cipher-suite gcm-aes-xpn-256
end
```

Apply the XPN MKA Policy to an Interface

To apply the XPN MKA policy to an interface, perform the following tasks:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface name*
4. **mka policy** *policy-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**configure terminal**
	Example:
	`router# configure terminal`
	`router(config)#`
		Enters global configuration mode.

Step 3	**interface** *interface name*
	Example:
	`router(config)# interface Te 0/1/0`
	`router(config-if)#`
		Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.

	Command or Action	Purpose
Step 4	**mka policy** *policy-name* **Example:** `router(config-if)# mka policy mka-xpn-policy`	Applies the XPN MKA protocol policy to the interface.
Step 5	**end** **Example:** `router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example

The following example shows the configuration to apply MACsec XPN policy to an interface:

```
enable
configure terminal
interface Te 0/1/0
mka policy mka-xpn-policy
end
```

Verifying MACsec eXtended Packet Numbering (XPN) Configuration

The following is a sample output of the **show mka session details** command with 128-bit XPN cipher suite configured. The SAK cipher suite output is indicated as GCM-AES-XPN-128.

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

Local Tx-SCI..... 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier..... 43
Interface Name..... Te0/1/0
Audit Session ID.....
CAK Name (CKN)..... 0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)..... 89572
EAP Role..... NA
Key Server..... YES
MKA Cipher Suite..... AES-128-CMAC

Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status..... FIRST-SAK
Old SAK AN..... 0
Old SAK KI (KN)..... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time..... 0s (No Old SAK to retire)

MKA Policy Name..... p2
Key Server Priority..... 2
Delay Protection..... NO
Replay Protection..... YES
```

```
Replay Window Size..... 0
Confidentiality Offset... 0
Algorithm Agility..... 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite..... 0080C20001000003 (GCM-AES-XPN-128)
MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired..... YES

# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 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 sample output of the **show mka session details** command with 256-bit XPN cipher suite configured. The SAK cipher suite output is indicated as GCM-AES-XPN-256.

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

Local Tx-SCI..... 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier..... 43
Interface Name..... Te0/1/0
Audit Session ID.....
CAK Name (CKN)..... 01000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)..... 89572
EAP Role..... NA
Key Server..... YES
MKA Cipher Suite..... AES-128-CMAC

Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status..... FIRST-SAK
Old SAK AN..... 0
Old SAK KI (KN)..... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time..... 0s (No Old SAK to retire)

MKA Policy Name..... p2
Key Server Priority..... 2
Delay Protection..... NO
Replay Protection..... YES
Replay Window Size..... 0
Confidentiality Offset... 0
Algorithm Agility..... 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite..... 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired..... YES
```

```
# of MACsec Capable Live Peers............ 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                       MN          Rx-SCI (Peer)        KS Priority
  ------------------------------------------------------------------------
  38046BA37D7DA77E06D006A9  89560       c800.8459.e764/002a   10
Potential Peers List:
  MI                       MN          Rx-SCI (Peer)        KS Priority
  ------------------------------------------------------------------------
Dormant Peers List:
  MI                       MN          Rx-SCI (Peer)        KS Priority
  ------------------------------------------------------------------------
```

The following is a sample output of the **show mka policy** command with 128-bit XPN cipher suite configured:

```
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                KS  DP  CO  SAKR   ICVIND   Cipher             Interfaces
Name                  Prio                         OLPL                  Suite(s)
   Applied
=============================================================================================================

*DEFAULT POLICY* 0     FALSE     0  FALSE     TRUE            GCM-AES-128          N/A
                                                                        GCM-AES-256

xpn128           0     FALSE     0  FALSE     TRUE                 GCM-AES-XPN-128
Te0/1/0
```

Configuring an Option to Change the EAPoL Ethernet Type

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **eapol** *eth-type*
5. **exit**

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
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** **Example:** `Device> enable`	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 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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 123456789012345678901234567890123456
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 12345678901234567890123456789012
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 12345678901234567890123456789012
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 12345678901234567890123456789012
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 be 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

!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
```

Example: Port-Channel Configuration with MACsec

The following is the sample configuration for port-channel configuration with MACsec on two seperate interfaces of a link bundle.

```
key chain kc1 macsec
  key 01
  key-string 12345678901234567890123456789012
  cryptographic-algorithm aes-128-cmac

key chain kc2 macsec
  key 02
  key-string 12345678901234567890123456789013
  cryptographic-algorithm aes-128-cmac

mka policy policy1
  macsec-cipher-suite gcm-aes-256

!Port-Channel Configuration

 interface Port-channel2
  mtu 9216
  ip mtu 9184
  ip address 10.3.1.3 255.255.255.0
  load-interval 30
  bfd interval 750 min_rx 750 multiplier 5
  lacp min-bundle 2
  no shut
  exit

!Member link configuration 1

 interface TenGigabitEthernet0/1/1
  no shut
```

```
  mtu 9216
  no ip address
  ip mtu 9184
  load-interval 30
  cdp enable
  no cdp tlv app
  mka policy policy1
  mka pre-shared-key key-chain kc1
  macsec
  lacp rate fast
  channel-group 2 mode active

!Member link configuration 2

  interface TenGigabitEthernet0/1/2
  no shut
  mtu 9216
  no ip address
  ip mtu 9184
  load-interval 30
  cdp enable
  no cdp tlv app
  mka policy policy1
  mka pre-shared-key key-chain kc2
  macsec
  lacp rate fast
  channel-group 2 mode active
```

Example: MACsec Fallback PSK Configuration

The **show mka sessions** command shows the global count of fallback sessions secured in the system, if any, as well as the status of the primary and fallback sessions.

The following output shows the primary key in Secured state, and the fallback key in Pending state:

```
router# show mka sessions
  Total MKA Sessions.........1
              Secured Sessions.....1
              Pending Sessions.....0
=============================================================================================

  Interface       Local_TxSCI            Policy-Name         Inherited        Key-Server

  Port-ID         Peer-RxSCI             MACsec-Peers        Status           CKN


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

 *Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    1                    Secured          02

       Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    1                    Pending          03
```

The following output shows that a key mismatch occurred in the primary key chain during rekeying, and that the fallback session is activated:

```
router# show mka sessions
  Total MKA Sessions.........2
              Secured Sessions.....1
```

```
              Pending Sessions.....0
=============================================================================================

  Interface       Local_TxSCI            Policy-Name         Inherited        Key-Server

  Port-ID         Peer-RxSCI             MACsec-Peers        Status           CKN


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

 *Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    1                   Rekeying         02

                                              0                   Init             07

       Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    1                   Secured          03
```

The following output shows primary session failure, the primary key session in Init state, and the fallback session in Secured state:

```
router# show mka sessions
  Total MKA Sessions.........1
              Secured Sessions.....0
              Pending Sessions.....1
=============================================================================================

  Interface       Local_TxSCI            Policy-Name         Inherited        Key-Server

  Port-ID         Peer-RxSCI             MACsec-Peers        Status           CKN


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

 *Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    0                   Init             07

       Te0/0/6         0c75.bd31.5c06.000e    POLICY              NO               YES
       14              78bc.1a60.c387/000f    1                   Secured          02
```

Troubleshooting MACsec Fallback PSK

The following syslog message indicates ICV verification failure on the primary key chain:

```
MKA-3-MKPDU_ICV_VERIFICATION_FAILURE
```

The following syslog message indicates that the session is secured with the fallback key chain:

```
MKA-5-SESSION_SECURED_FALLBACK
```

Additional References

Related Documents

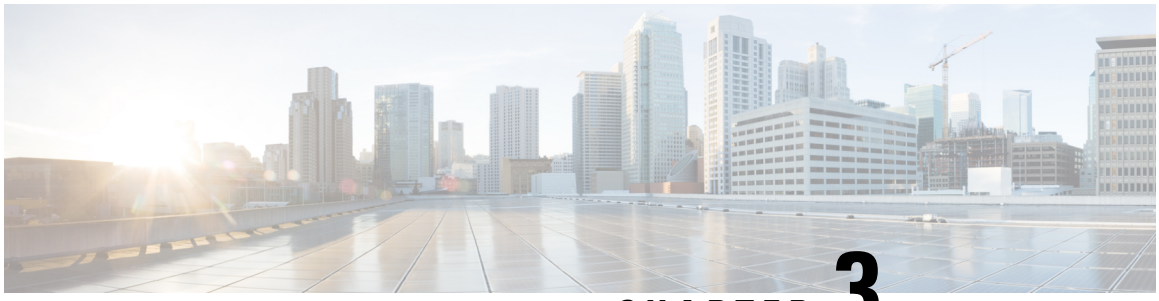
Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

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

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



CHAPTER 3

MACsec Smart Licensing

MACsec Smart Licensing Overview

This chapter provides an overview of MACsec Smart Licensing. Smart Licensing feature is a standardized licensing platform that simplifies the Cisco software experience and helps you to understand how Cisco software is used across your network. Smart Licensing is the next generation licensing platform for all Cisco software products. MACsec licensing allows you to enable CSL permanent and Smart Licensing on Cisco ASR 1000 platforms.

Feature Information for MACsec Smart Licensing

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 7: Feature Information for MACsec Smart Licensing

Feature Name	Releases	Feature Information
MACsec and DLC Support	Cisco IOS XE Fuji 16.9.1	Smart Licensing feature is a standardized licensing platform that simplifies the Cisco software experience and helps you to understand how Cisco software is used across your network. Smart Licensing is the next generation licensing platform for all Cisco software products. No commands were introduced or modified by this feature.

Information about MACsec Smart Licensing

Effective with Cisco IOS XE Fuji Release 16.9.1, MACsec Smart Licensing (SL) is supported on the following platforms:

Ports	License Feature	License PID	Supported Platform		
			MIP-100 (RP2/RP3)	ASR1001-HX	ASR1002-HX
BUILT-IN 1 GE ports	MACSEC1G	FLSA1-MACSEC1G	N/A	Yes	Yes
BUILT-IN 10 GE ports	MACSEC10G	FLSA1-MACSEC10G	N/A	Yes	Yes
EPA-18X1GE	MACSEC1G	FLSA1-MACSEC1G	Yes	N/A	Yes
EPA-10X10GE	MACSEC10G	FLSA1-MACSEC10G	Yes	N/A	Yes
EPA-1X40GE	MACSEC40G	FLSA1-MACSEC40G	Yes	N/A	Yes
EPA-2X40GE	MACSEC40G	FLSA1-MACSEC40G	Yes	N/A	Yes
EPA-QSFP-1X100GE	MACSEC100G	FLSA1-MACSEC100G	Yes	N/A	Yes

MACsec licenses are available for each port and are applicable only for physical ports (sub-interfaces do not require additional license). Device Led Conversion (DLC) support is available for MACsec port licenses to ensure that your paper licenses are added to smart account.

The Device-led conversion allows license migration from Classic to Smart license automatically for licences that are on the devices. The devices needs to be registered in Cisco Smart Software Manager (SSM) for automatic conversion to smart license.

Note

- According to earlier releases, ASR1001-X built-in continues with the IPsec license which acts as MACsec license.
- MACsec license is not supported for EPA-1X100GE and EPA-CPAK-2X40GE.
- CSL – EvalRTU license is not available for MACsec licenses.

One unit of MACsec license is used when a port containing MACsec configuration is unshut or when the configuration is applied on an unshut port.

One unit of MACsec license is released when a port containing MACsec configuration is shut or when the configuration is removed from an unshut port.

Deployment and Migration Examples

MACSec support is available in Cisco Software License (CSL) and Smart License (SL) modes from Cisco IOS XE Fuji 16.9.1. However, for releases after 16.9.1, MACSec will support only Smart License.

The following scenarios explain how an existing router is deployed and migrated to Cisco IOS XE Fuji 16.9.1:

Upgrade in CSL Mode when a permanent license is installed

If MACsec permanent licenses are installed on the device before upgrading (prior to Cisco IOS XE Fuji 16.9.1 release), then these licenses are used after the upgrade.

- Before the upgrade, assume the following:
  - Router is operating on a release prior to Cisco IOS XE Fuji 16.9.1
  - MACsec is configured on four un-shut 1G interfaces
  - Four MACSEC1G permanent licenses are installed
- After the upgrade, four MACSEC1G licenses are used.

Upgrade in CSL Mode when a permanent license is not installed

When MACsec is configured on unshut ports, ideally EvalRTU licenses should be used after the upgrade. Since EvalRTU support is not available, the license request is skipped and a warning message is displayed. For example:

%IOSXE_LICENSE_POLICY_MANAGER-4-INSUF_PERM_LIC: 0/0/0: Insufficient MACSEC40G permanent license, skipping license request assuming customer has honour license

- Before the upgrade, assume the following:
  - Router is operating on a release prior to Cisco IOS XE Fuji 16.9.1
  - MACsec is configured on four un-shut 1G interfaces
- After the upgrade
  - No MACsec license is used

- Warning message is displayed
- If you install four permanent licenses at a later point of time, then these licenses are used immediately

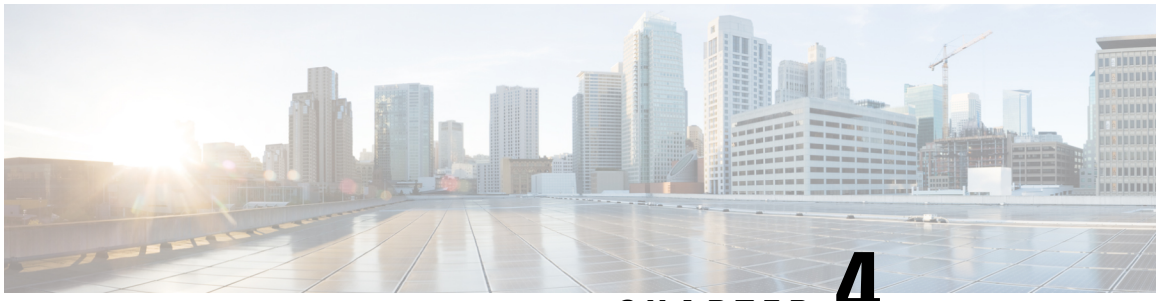
Migration to SL Mode

To avoid **Out of Compliance** scenario, all Product Activation Keys (PAK) and non-PAK licenses should be added to customer's virtual CSSM account.

The Device Led Conversion (DLC) feature migrates licenses to Smart Account. For DLC to work properly, all licenses should be enabled in CSL mode before migrating to SL mode.

Perform the following steps to migrate to SL Mode:

- Upgrade from releases prior to Cisco IOS XE 16.9.1 to Cisco IOS XE 16.9.1
  1. Upgrade to Cisco IOS XE Fuji 16.9.1 in CSL mode
  2. Migrate to SL mode and trigger DLC
- Upgrade from releases prior to Cisco IOS XE Fuji 16.9.1 to later releases
  1. Upgrade to Cisco IOS XE Fuji 16.9.1 in CSL mode
  2. Migrate to SL mode and trigger DLC
  3. Upgrade to releases later than Cisco IOS XE Fuji 16.9.1



CHAPTER 4

Certificate-based MACsec Encryption

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

Certificate-based MACsec encryption can be done using either remote authentication or local authentication.

Feature Information for Certificate-based MACsec Encryption

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 8: Feature Information for Certificate-based MACsec Encryption

Feature Name	Releases	Feature Information
Certificate-based MACsec Encryption	Cisco IOS XE Everest Release 16.6.1	The Certificate-based MACsec Encryption feature uses 802.1X port-based authentication with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) to carry Certificates for router ports where MACsec encryption is required. EAP-TLS mechanism is used to do the mutual authentication and to get the Primary Session Key from which the Connectivity Association Key (CAK) is derived for the MACsec Key Agreement (MKA) protocol.

Prerequisites for Certificate-based MACsec Encryption

- 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. Refer to the *Cisco Identity Services Engine Administrator Guide, Release 2.3*.
- 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.

Restrictions for Certificate-based MACsec Encryption

- MKA is not supported on port-channels.
- High Availability for MKA is not supported.
- Certificate-based MACsec encryption on sub-interfaces is not supported.

Information About Certificate-based MACsec Encryption

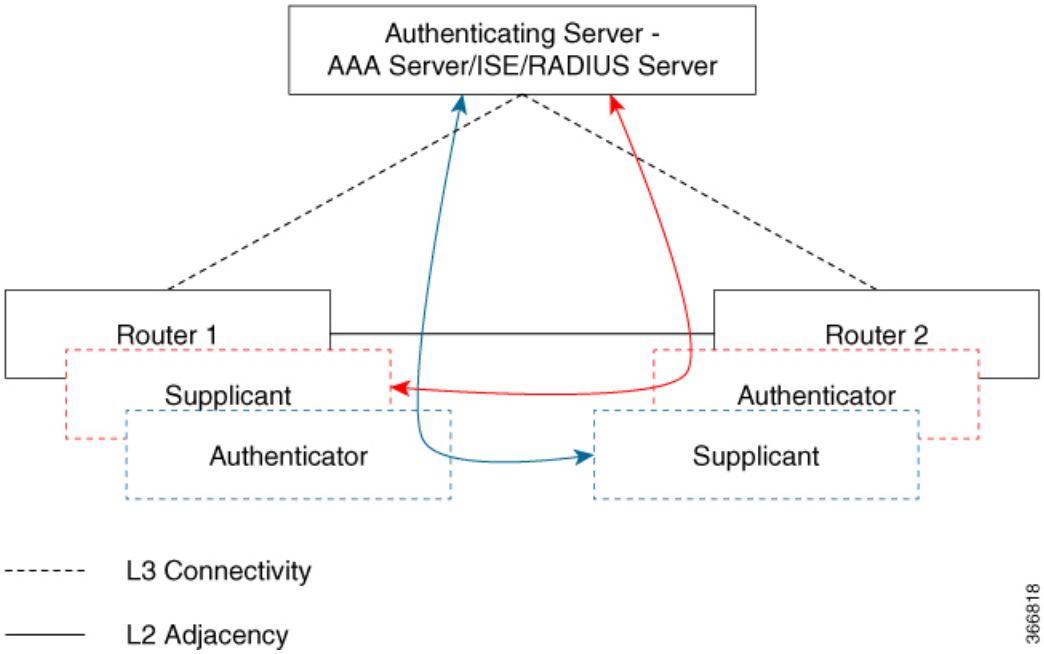
MKA MACsec is supported on router-to-router links. Using IEEE 802.1X Port-based Authentication with Extensible Authentication Protocol (EAP-TLS), you can configure MKA MACsec between device ports. EAP-TLS allows mutual authentication and obtains an primary session key from which the connectivity association key (CAK) is derived for MKA protocol. Device certificates are carried, using EAP-TLS, for authentication to the AAA server.

Call Flow for Certificate-based MACsec Encryption using Remote Authentication

Suppllicants are unauthorized devices that try to gain access to the network. Authenticators are devices that control the physical access to the network based on the authentication status of the supplicant.

As shown in the following diagram, the devices are connected directly. The router acts as both EAP Supplicant and Authenticator on the port.

The figure below depicts two EAP call flows (with separate EAP-Session ID) on the router. The red flow depicts Router 1 as supplicant and Router 2 as authenticator and the blue flow is vice-versa.



When the interface is configured for 802.1x role as both, The authentication manager on a router creates a session with two EAP session (blue and red with separate EAP session ID) flows with supplicant as well as an authenticator role and both trigger EAP-TLS mutual authentication with the remote authenticating server (AAA server/ISE/RADIUS).

After mutual authentication, the MSK of the flow corresponding to the router with the higher MAC address and role as authenticator is picked to derive the CAK.

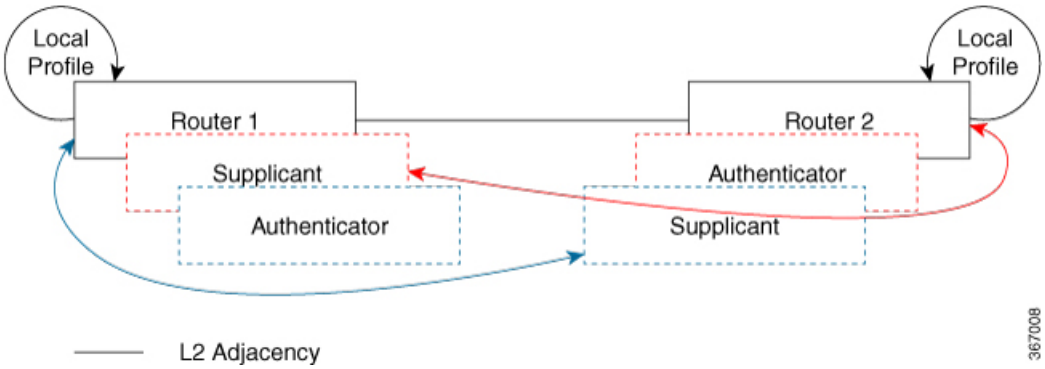
In the diagram above, if Router 1 MAC address is less than Router 2, then the primary session key (PSK) obtained from the EAP session (blue flow) is used as EAP-PSK for the MKA (Router 1 acts as authenticator and Router 2 as supplicant). This ensures that Router 1 acts as MKA Key Server and Router 2 is the Non-Key Server.

If the Router 2 MAC Address is less than Router 1 then the PSK obtained from the EAP session (red flow) is used (by both routers) as EAP-PSK for the MKA to derive the CAK.

Call Flow for Certificate-based MACsec Encryption using Local Authentication

As shown in the following diagram, the devices are connected directly. The router acts as both EAP Supplicant and Authenticator on the port.

The figure below depicts two EAP call flows (with separate EAP-Session ID) on the router. The red flow depicts Router 1 as supplicant and Router 2 as authenticator and the blue flow is vice-versa.



When the interface is configured for 802.1x role as both, The authentication manager on a router creates a session with two EAP session (blue and red with separate EAP session ID) flows with supplicant as well as an authenticator role and both trigger EAP-TLS mutual authentication with the local authenticating server.

After mutual authentication, the PSK of the flow corresponding to the router with the higher MAC address and role as authenticator is picked to derive the CAK.

In the diagram above, if Router 1 MAC address is less than Router 2, then the primary session key (PSK) obtained from the EAP session (blue flow) is used as EAP-PSK for the MKA (Router 1 acts as authenticator and Router 2 as supplicant). This ensures that Router 1 acts as MKA Key Server and Router 2 is the Non-Key Server.

If the Router 2 MAC Address is less than Router 1 then the PSK obtained from the EAP session (red flow) is used (by both routers) as EAP-PSK for the MKA to derive the CAK.

Configuring Certificate-based MACsec Encryption using Remote Authentication

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

Configuring Certificate Enrollment

Generating Key Pairs

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto key generate rsa label** *label name* **general-keys modulus** *size*	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 4	**end**	Returns to privileged EXEC mode.
Step 5	**show authentication session interface** *interface-id*	Verifies the authorized session security status.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

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**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	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** *label*	Specifies which key pair to associate with the certificate. **Note** The **rsakeypair** name must match the trust-point name.

	Command or Action	Purpose
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**auto-enroll** *percent* **regenerate**	Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA. If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration. By default, only the Domain Name System (DNS) name of the device is included in the certificate. Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will 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	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**exit**	Exits global configuration mode.
Step 12	**show crypto pki certificate** *trustpoint name*	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**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	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** *label*	Specifies which key pair to associate with the certificate.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**exit**	Exits Global Configuration mode.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**crypto pki enroll** *name*	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. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate**	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.

	Command or Action	Purpose
		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	**exit**	Exits Global Configuration mode.
Step 14	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.
Step 15	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Enabling 802.1x Authentication and Configuring AAA

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**dot1x system-auth-control**	Enables 802.1X on your device.
Step 5	**radius server** *name*	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	**address** *ip-address* **auth-port** *port-number* **acct-port** *port-number*	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 7	**automate-tester username** *username*	Enables the automated testing feature for the RADIUS server. With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks for a RADIUS response from the server. A success message is not necessary - a failed authentication suffices, because it shows that the server is alive.
Step 8	**key** *string*	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.

	Command or Action	Purpose
Step 9	**radius-server deadtime** *minutes*	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
Step 10	**exit**	Returns to global configuration mode.
Step 11	**aaa group server radius** *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
Step 12	**server** *name*	Assigns the RADIUS server name.
Step 13	**exit**	Returns to global configuration mode.
Step 14	**aaa authentication dot1x default group** *group-name*	Sets the default authentication server group for IEEE 802.1x.
Step 15	**aaa authorization network default group** *group-name*	Sets the network authorization default group.

Configuring EAP-TLS Profile and 802.1x Credentials

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**eap profile** *profile-name*	Configures EAP profile and enters EAP profile configuration mode.
Step 4	**method tls**	Enables EAP-TLS method on the device.
Step 5	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**dot1x credentials** *profile-name*	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	**username** *username*	Sets the authentication user ID.
Step 9	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 10	**end**	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 4	**macsec**	Enables MACsec on the interface.
Step 5	**authentication periodic**	Enables reauthentication for this port.
Step 6	**authentication timer reauthenticate interval**	Sets the reauthentication interval.
Step 7	**access-session host-mode multi-domain**	Allows hosts to gain access to the interface.
Step 8	**access-session closed**	Prevents preauthentication access on the interface.
Step 9	**access-session port-control auto**	Sets the authorization state of a port.
Step 10	**dot1x pae both**	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	**dot1x credentials profile**	Assigns a 802.1x credentials profile to the interface.
Step 12	**dot1x supplicant eap profile** *name*	Assigns the EAP-TLS profile to the interface.
Step 13	**service-policy type control subscriber** *control-policy name*	Applies a subscriber control policy to the interface.
Step 14	**exit**	Returns to privileged EXEC mode.
Step 15	**show macsec interface**	Displays MACsec details for the interface.
Step 16	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Certificate-based MACsec Encryption using Local Authentication

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

Configuring the EAP Credentials using Local Authentication

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**aaa local authentication default authorization default**	Sets the default local authentication and default local authorization method.
Step 5	**aaa authentication dot1x default local**	Sets the default local username authentication list for IEEE 802.1x.
Step 6	**aaa authorization network default local**	Sets an authorization method list for local user.
Step 7	**aaa authorization credential-download default local**	Sets an authorization method list for use of local credentials.
Step 8	**exit**	Returns to privileged EXEC mode.

Configuring the Local EAP-TLS Authentication and Authorization Profile

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**aaa new-model**	Enables AAA.
Step 4	**dot1x credentials** *profile-name*	Configures the dot1x credentials profile and enters dot1x credentials configuration mode.
Step 5	**username** *name* **password** *password*	Sets the authentication user ID and password.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**aaa attribute list** *list-name*	(Optional) Sets the AAA attribute list definition and enters attribute list configuration mode.
Step 8	**aaa attribute type linksec-policy must-secure**	(Optional) Specifies the AAA attribute type.
Step 9	**exit**	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	**username** *name* **aaa attribute list** *name*	(Optional) Specifies the AAA attribute list for the user ID.
Step 11	**end**	Returns to privileged EXEC mode.

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**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	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** *label*	Specifies which key pair to associate with the certificate. **Note** The **rsakeypair** name must match the trust-point name.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**auto-enroll** *percent* **regenerate**	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.

	Command or Action	Purpose
		By default, only the Domain Name System (DNS) name of the device is included in the certificate. Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will 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	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**exit**	Exits global configuration mode.
Step 12	**show crypto pki certificate** *trustpoint name*	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**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *server name*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem*	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.

	Command or Action	Purpose
Step 5	**rsakeypair** *label*	Specifies which key pair to associate with the certificate.
Step 6	**serial-number none**	The **none** keyword specifies that a serial number will not be included in the certificate request.
Step 7	**ip-address none**	The **none** keyword specifies that no IP address should be included in the certificate request.
Step 8	**revocation-check crl**	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	**exit**	Exits Global Configuration mode.
Step 10	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
Step 11	**crypto pki enroll** *name*	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. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate**	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	**exit**	Exits Global Configuration mode.
Step 14	**show crypto pki certificate** *trustpoint name*	Displays information about the certificate for the trust point.

	Command or Action	Purpose
Step 15	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring EAP-TLS Profile and 802.1x Credentials

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**eap profile** *profile-name*	Configures EAP profile and enters EAP profile configuration mode.
Step 4	**method tls**	Enables EAP-TLS method on the device.
Step 5	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**dot1x credentials** *profile-name*	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	**username** *username*	Sets the authentication user ID.
Step 9	**pki-trustpoint** *name*	Sets the default PKI trustpoint.
Step 10	**end**	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.

	Command or Action	Purpose
Step 4	**macsec**	Enables MACsec on the interface.
Step 5	**authentication periodic**	Enables reauthentication for this port.
Step 6	**authentication timer reauthenticate interval**	Sets the reauthentication interval.
Step 7	**access-session host-mode multi-domain**	Allows hosts to gain access to the interface.
Step 8	**access-session closed**	Prevents preauthentication access on the interface.
Step 9	**access-session port-control auto**	Sets the authorization state of a port.
Step 10	**dot1x pae both**	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	**dot1x credentials profile**	Assigns a 802.1x credentials profile to the interface.
Step 12	**dot1x authenticator eap profile** *name*	Assigns the EAP-TLS authenticator profile to the interface.
Step 13	**dot1x supplicant eap profile** *name*	Assigns the EAP-TLS supplicant profile to the interface.
Step 14	**service-policy type control subscriber** *control-policy name*	Applies a subscriber control policy to the interface.
Step 15	**exit**	Returns to privileged EXEC mode.
Step 16	**show macsec interface**	Displays MACsec details for the interface.
Step 17	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Verifying Certificate-based MACsec Encryption

Use the following **show** commands to verify the configuration of certificate-based MACsec encryption. Given below are the sample outputs of the **show** comamnds.

The **show mka sessions** command displays a summary of active MACsec Key Agreement (MKA) Protocol sessions.

```
Device# show mka sessions

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

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

Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN

=============================================================================================
Te0/1/3        74a2.e625.4413/0013 *DEFAULT POLICY* NO                YES
```

```
19            74a2.e625.4c22/0012 1               Secured
10000000000000000000000000000000000000000000000000000000000000000000
```

The **show macsec status interface** *interface-id* displays MACsec status information for the given interface.

```
Device# show macsec status interface te0/1/2

  Capabilities:
  Ciphers Supported:        GCM-AES-128 GCM-AES-256
  Cipher:                   GCM-AES-128
  Confidentiality Offset:   0
  Replay Window:            64
  Delay Protect Enable:     FALSE
  Access Control:           must-secure

Transmit SC:
  SCI:                      74A2E6254C220012
  Transmitting:             TRUE
Transmit SA:
  Next PN:                  412
  Delay Protect AN/nextPN:  99/0

Receive SC:
  SCI:                      74A2E62544130013
  Receiving:                TRUE
Receive SA:
  Next PN:                  64
  AN:                       0
  Delay Protect AN/LPN:     0/0
```

The **show access-session interface** *interface-id* **details** displays detailed information about the access session for the given interface.

```
Device# show access-session interface te1/0/1 details

   Interface:  TenGigabitEthernet1/0/1
             IIF-ID:  0x17298FCD
         MAC Address:  f8a5.c592.13e4
        IPv6 Address:  Unknown
        IPv4 Address:  Unknown
           User-Name:  DOT1XCRED
              Status:  Authorized
              Domain:  DATA
      Oper host mode:  multi-host
    Oper control dir:  both
     Session timeout:  N/A
   Common Session ID:  000000000000000BB72E8AFA
     Acct Session ID:  Unknown
              Handle:  0xc3000001
      Current Policy:  MUSTS_1


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

Server Policies:


Method status list:
```

```
      Method          State
    dot1xSup          Authc Success
       dot1x          Authc Success
```

Configuration Examples for Certificate-based MACsec Encryption

Example: Enrolling the Certificate

```
Configure Crypto PKI Trustpoint:
  crypto pki trustpoint POLESTAR-IOS-CA
   enrollment terminal
   subject-name CN=ASR1000x1@polestar.com, C=IN, ST=KA, OU=ENG,O=Polestar
   revocation-check none
   rsakeypair mkaioscarsa
   storage nvram:
  !
Manual Installation of Root CA certificate:
 crypto pki authenticate POLESTAR-IOS-CA
```

Example: Enabling 802.1x Authentication and AAA Configuration

```
aaa new-model
dot1x system-auth-control
radius server ISE
 address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646
 automate-tester username dummy
 key dummy123
 radius-server deadtime 2
!
aaa group server radius ISEGRP
 server name ISE
!
aaa authentication dot1x default group ISEGRP
aaa authorization network default group ISEGRP
```

Example: Configuring EAP-TLS Profile and 802.1X Credentials

```
eap profile EAPTLS-PROF-IOSCA
 method tls
 pki-trustpoint POLESTAR-IOS-CA
!

dot1x credentials EAPTLSCRED-IOSCA
 username asr1000@polestar.company.com
 pki-trustpoint POLESTAR-IOS-CA
!
```

Example: Applying 802.1X, PKI, and MACsec Configuration on the Interface

```
interface TenGigabitEthernet0/1
 macsec network-link
 authentication periodic
 authentication timer reauthenticate <reauthentication interval>
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae both
 dot1x credentials EAPTLSCRED-IOSCA
 dot1x supplicant eap profile EAPTLS-PROF-IOSCA
 service-policy type control subscriber DOT1X_POLICY_RADIUS
```

Additional References

Related Documents

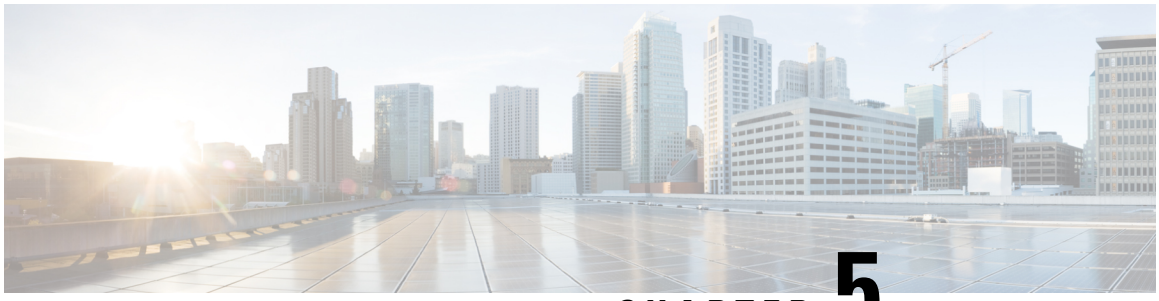
Related Topic	Document Title
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

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

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



CHAPTER 5

MACsec as a Service-An Encryption Solution

This document describes how to deploy an encryption solution - Cisco MACsec as a Service, to secure network traffic using Cisco WAN MACsec and Ethernet Virtual Circuit (EVC). This solution provides Ethernet Virtual Circuit (EVC) support for Media Access Control security (MACsec) with MACsec Key Agreement (MKA) protocol. MACsec with MKA detects EVCs and enables the physical interface that matches the EVC criteria. With this functionality, users can transport layer2 traffic from multiple enterprises over a WAN link and independently secure their traffic with MACsec over EVC.

Feature Information for MACsec as a Service

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 9: Feature Information for MACsec as a Service

Feature Name	Releases	Feature Information
MACsec as a Service - Ethernet Virtual Circuit Support for MACsec and MKA	Cisco IOS XE Gibraltar 16.12.1a	This document describes how to deploy an encryption solution using Ethernet Virtual Circuit (EVC) support for MACsec with MACsec Key Agreement (MKA) protocol. MACsec with MKA detects EVCs and enables the physical interface that matches the EVC criteria. With this functionality, users can transport layer 2 traffic from multiple enterprises over a WAN link and independently secure their traffic with MACsec over EVC. In this release, the feature is supported only on Cisco ASR1000 Series Aggregation Services Routers. The following commands were introduced or modified: **mka pre-shared-key key-chain** *key-chain-name*, **mka policy** *policy-name*, **mka default-policy**, **macsec replay-protection window** *window size*, **eapol destination-address** *destination-address*{*bridge-group-address* | *broadcast-address* | *lldp-multicast-address* | *unicast mac-address*}, **eapol eth-type** *eth-type* .

Prerequisites for Ethernet Virtual Circuit Support for MACsec and MKA

- WAN MACsec requires a MACsec license. See the Table in Cisco ASR 1000 Series Ethernet Line Cards Datasheet
- Ensure that the Layer2 transparent Ethernet Services are available. The service provider network must provide a MACsec Layer2 Control Protocol transparency, such as, Extensible Authentication Protocol over LAN (EAPoL).

Restrictions for Ethernet Virtual Circuit Support for MACsec and MKA

- This feature is supported only on Cisco 1000 Series Aggregation Services Routers.
- This feature is supported from Cisco IOS XE Gibraltar 16.12.1a.
- Only dot1q based header is supported on EVC with MACsec.

  Number of MKA P2P sessions per port is 8 on 1 Gig and 32 on 10 Gig interfaces.
- If MACsec or MKA session is already configured on a physical interface or on a sub-interface, then you cannot configure MACsec with MKA session under the service instance or EVC mode on the same physical interface and vice versa.
- MACsec EVC is supported only with MKA PSK based sessions.

Information About Ethernet Virtual Circuit Support for MACsec and MKA

MACsec and MKA Overview

MACsec is an IEEE 802.1AE standard 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 MKA is supported on downlink ports for encryption between the routers or switches and host devices. MKA is the control plane for MACsec, which is defined in the IEEE standard 802.1X. MKA frames form part of the EAPoL frames. MACsec is the last mile in the packet processing path and encrypts all the traffic except the EAPoL frames.

For implementing WAN MACsec and MKA, verify that a basic Layer 2 Ethernet connectivity is established before attempting to enable MACsec. For more information, refer to the MACsec and MKA Overview section.

Cisco Ethernet Virtual Circuit

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service. It embodies the different parameters on which the service is being offered. In the Cisco EVC structure, the bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given router. Service instance is associated with a bridge domain (BD) based on the configuration.

An incoming frame can be classified as service instance based on the following criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Service instance also supports the alternative mapping criteria:
- Untagged—Mapping to all the frames lacking a 802.1Q or 802.1ad header
- Default—Mapping to all the frames

For more information on the EVC architecture, see "Configuring Ethernet Virtual Circuit" section on the in the Carrier Ethernet Configuration guide.

Ethernet Service Instance or Ethernet Flow Point

Ethernet Flow Point (EFP) is a transport-agnostic abstraction of an Ethernet service on an interface. It classifies frames from a same physical port to one of the multiple service instances associated with the port based on the user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MACsec Key Agreement (MKA) is used as the control protocol. MKA selects the cipher suite, which is used for encryption and exchanges 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 on either physical or on a 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.

Bridge Domain (BD) defines a broadcast domain internal to the platform and it allows decoupling broadcast domain from VLAN thus enables per-port VLAN significance. This removes the scalability limitations associated with a single per-box VLAN ID space. For more information on how EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP), refer to Bridge Domain Interface Encapsulation.

Benefits of MACsec and MKA with Ethernet Virtual Circuit

- Transport the Layer2 VLANs from multiple enterprise customers over a WAN link and independently secure their traffic using MACsec.

  Selective encryption of the LAN traffic over WAN using MACsec

For more information on the benefits of WAN MACsec and MKA Support, refer to the Benefits of WAN MACsec and MKA Support Enhancements section.

MACsec as a Service using Ethernet Virtual Circuit

The topologies below describe how to deploy Ethernet Virtual Circuit (EVC) with WAN MACsec in an EoMPLS network in a Point-to-Point and Point to Multi-Point scenarios. The traffic, which is encrypted, flows from CEs with CVLAN to the CE routers, and the CE routers in the network ensure that the data reaches their destination.

Figure 1: MKA and MACsec Topology with a single SVLAN

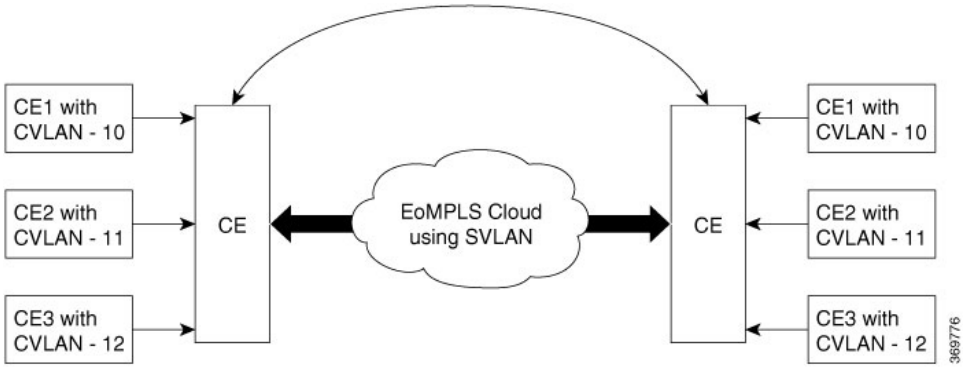
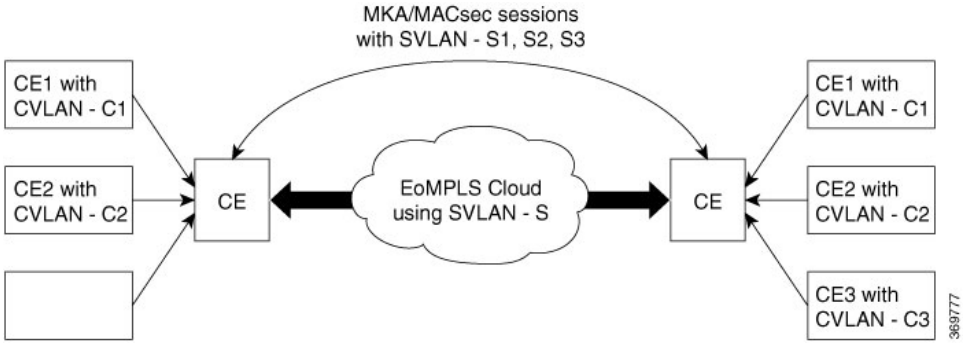


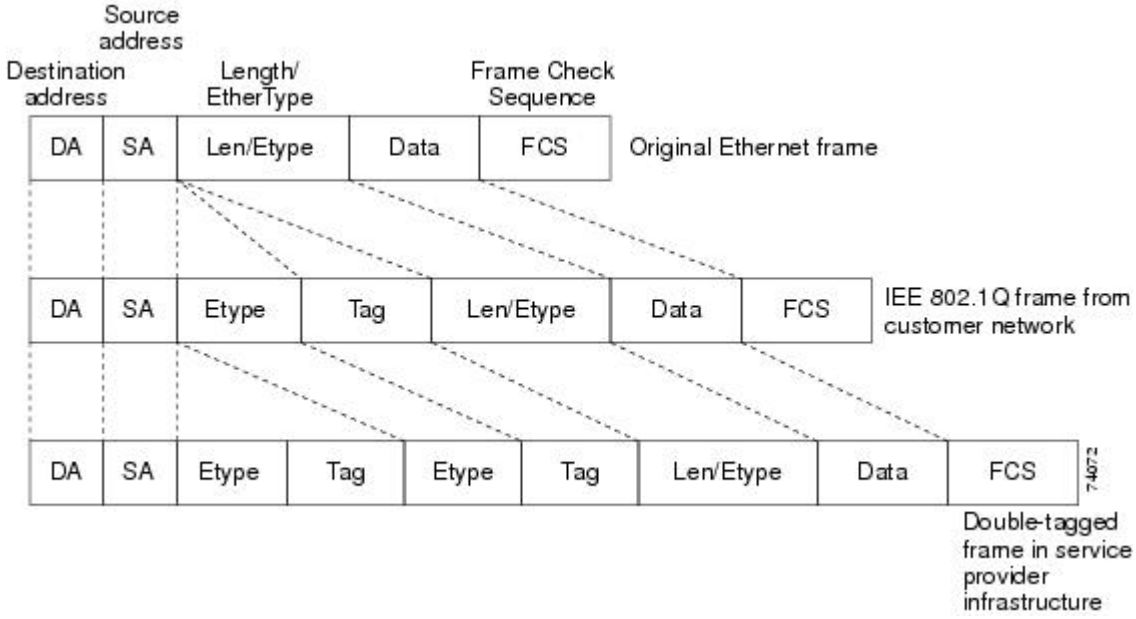
Figure 2: MKA and MACsec Topology with Multiple SVLANs



Cisco WAN MACsec, which supports EAPoL frames, not only encrypts the data, but, helps to seamlessly navigate across a diverse service provider network to securely connect all your remote sites.

In an EoMPLS network, you can connect multiple Layer 2 Ethernet networks at different locations. To enable connecting to different service providers over EoMPLS, WAN MACsec supports dot.1q tag in the clear, which helps connect to remote sites over public E-LINE or E-LAN services without disrupting the service provider network.

Figure 3: 802.1Q, and Double-Tagged Ethernet Packet Formats



Service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

When you use a service provider network to exchange data between networks, the EVC with MACsec helps to encrypt the data in transit. The dot.1q tag in clear opens a multitude of design options for securing complex networks. Using the EVCs, service providers can encapsulate packets that enter the service-provider network with multiple customer VLAN IDs (C-VLANs) and a single 0x8100 Ethertype VLAN tag with a service provider VLAN (S-VLAN). Within the service provider network, packets are switched based on the S-VLAN. When the packets egress the service provider network onto the customer network, the S-VLAN tag is decapsulated and the original customer packet is restored.

How to Configure Ethernet Virtual Circuit Support for MACsec and MKA

Configure Key Chain

To configure a key chain, perform the steps below:

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

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 keychain configuration mode

Step 4 **key** *hex-string*

Example:

```
Device(config-keychain)# key 01
```

Configures a key and enters keychain key configuration mode.

Step 5 **cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**}

Example:

```
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
```

Set cryptographic authentication algorithm.

Step 6 **key-string** *pwd-string*}

Example:

```
Device(config-keychain-key)# key-string 12345678901234567890123456789013
```

Sets the password for a key string.

Step 7 **end**

Example:

```
Device(config-keychain-key)# end
```

Returns to privileged EXEC mode.

Configure MKA and MACsec on Interfaces

To configure MKA and MACsec on an interface, perform these steps:

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters the configuration mode

Step 3 **mka policy** *policy-name*

Example:

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

Configures an MKA policy

Step 4 **mka pre-shared-key key-chain** *key-chain-name*

Example:

```
Device(config)# mka pre-shared-key key-chain 10
```

Configures an MKA pre-shared-key key-chain 10

Note The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.

Step 5 **macsec**

Configures MACsec for the EAPOL frame type.

Step 6 **macsec replay-protection window** *window-size*

Changes the replay window 10

Step 7 **end**

Returns to privileged EXEC mode.

Configure Ethernet Virtual Circuit on Ingress Port Facing Customer Edge

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Enters global configuration mode.

Step 3 **interface GigabitEthernet0/0/2**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 4 **service instance 10 Ethernet**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 5 **configure terminal**

Enters global configuration mode.

Step 6 **interface GigabitEthernet0/0/2**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 7 **encapsulation dot1q 10**

Step 8 **rewrite ingress tag push dot1q 20 symmetric**

Step 9 **bridge-domain** *number*

Step 10

```
interface GigabitEthernet0/0/2
   service instance 11 Ethernet
encapsulation dot1q 10
rewrite ingress tag push dot1q 20 symmetric
bridge-domain 21
interface GigabitEthernet0/0/2
   service instance 12 Ethernet
encapsulation dot1q 10
rewrite ingress tag push dot1q 20 symmetric
bridge-domain 22
```

Configure MACsec EVC on Egress Port Facing Service Provider Network

Step 1 **enable**

Step 2 **configure terminal**

Example:

```
interface tenGigabitEthernet0/1/1
    macsec dot1q-in-clear 1
    service instance 20 Ethernet
encapsulation dot1q 20
mka pre-shared-key key-chain kc1
macsec
bridge-domain 20
    service instance 21 Ethernet
encapsulation dot1q 21
mka pre-shared-key key-chain kc1
     macsec
     bridge-domain 21
    service instance 22 Ethernet
```

```
encapsulation dot1q 22
mka pre-shared-key key-chain kc1
macsec
bridge-domain 22
```

Verify Enablement of Pre-Shared-Key based on a Macsec and MKA session

SUMMARY STEPS

1. **enable**
2.

DETAILED STEPS

Step 1 **enable**

Step 2 **Example:**

```
show running-config | sec kc1
key chain kc1 macsec
 key 01
   cryptographic-algorithm aes-128-cmac
  key-string 123456789012345678901234567890123456
 mka pre-shared-key key-chain kc1
  mka pre-shared key-chain kc1
```

The following is sample configuration for enabling Pre-Shared-Key (PSK) based MKA/Macsec session with default policy under service instance mode:

```
Device#show running-config interface gi0/0/0
Building configuration...
 …
 …
 …
Current configuration : 142 bytes
!
interface Ethernet0/0
  no ip address
  negotiation auto
  service instance 10 ethernet
    encapsulation dot1q 10
    rewrite ingress tag pop 1 symmetric
    mka pre-shared key-chain kc1
    macsec
    bridge-domain 100
 !
end
```

Configuration Examples for Ethernet Virtual Circuit Support for MACsec and MKA

Example: General Troubleshooting

Example: General Troubleshooting

Example: Show MKA Configured Command

Example: Show MKA Configured Command

Example: Show Statistics

```
MACsec statistics on an EFP: To validate MACsec Statistics on an EFP instance, use
show macsec statistics interface gi0/0/3 efp 10
------------------------------------------------------------
MACsec Statistics for Gi0/0/3.EFP10
SecY Counters
  Ingress Untag Pkts:       5
  Ingress No Tag Pkts:      63440
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
  Ingress No SCI Pkts:      0
  Ingress Overrun Pkts:     0
  Ingress Validated Octets: 0
  Ingress Decrypted Octets: 0
  Egress Untag Pkts:        0
  Egress Too Long Pkts:     0
  Egress Protected Octets:  0
  Egress Encrypted Octets:  0
Controlled Port Counters
  IF In Octets:             0
  IF In Packets:            0
  IF In Discard:            63440
  IF In Errors:             0
  IF Out Octets:            0
  IF Out Packets:           0
  IF Out Errors:            0
 Transmit SC Counters (SCI: 70708BBA4683000A)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       0
Transmit SA Counters (AN 2)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       0
Receive SA Counters (SCI: 70708BBA4183000A  AN 2)
  In Pkts Unchecked:        0
  In Pkts Delayed:          0
  In Pkts OK:               0
  In Pkts Invalid:          0
```

```
In Pkts Not Valid:         0
In Pkts Not using SA:      0
In Pkts Unused SA:         0
In Pkts Late:              0
```

Example: Show efp commands

Example: Show efp commands

Additional References for Ethernet Virtual Circuit Support for MACsec and MKA

Related Documents

Standards and RFCs

Standard/RFC	Title
Standard	*Title*

MIBs

MIB	MIBs Link
• CISCO-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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/cisco/web/support/index.html