

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, on page 1
- Prerequisites for Ethernet Virtual Circuit Support for MACsec and MKA, on page 2
- Restrictions for Ethernet Virtual Circuit Support for MACsec and MKA, on page 2
- Information About Ethernet Virtual Circuit Support for MACsec and MKA, on page 3
- How to Configure Ethernet Virtual Circuit Support for MACsec and MKA, on page 6
- Configuration Examples for Ethernet Virtual Circuit Support for MACsec and MKA, on page 11
- Additional References for Ethernet Virtual Circuit Support for MACsec and MKA, on page 12

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 1: 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

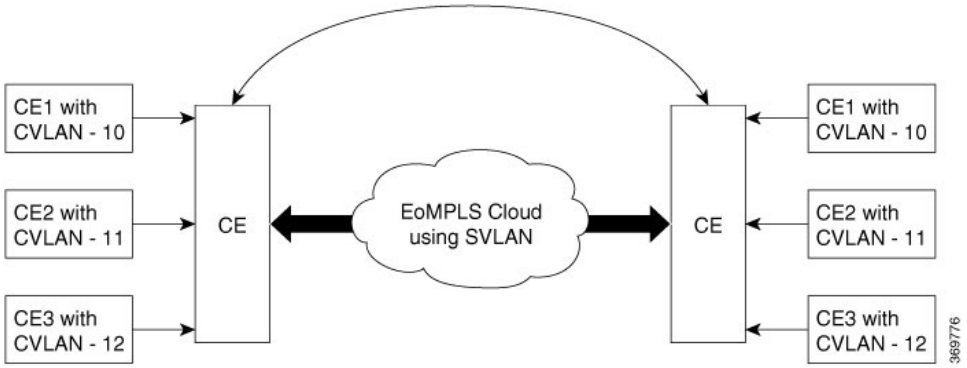
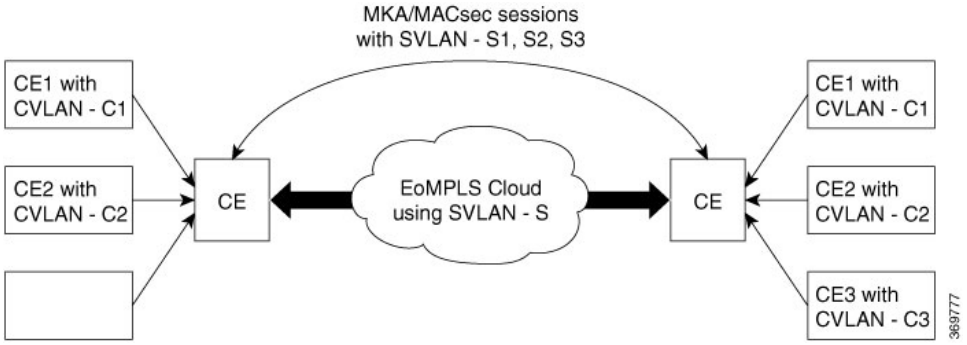


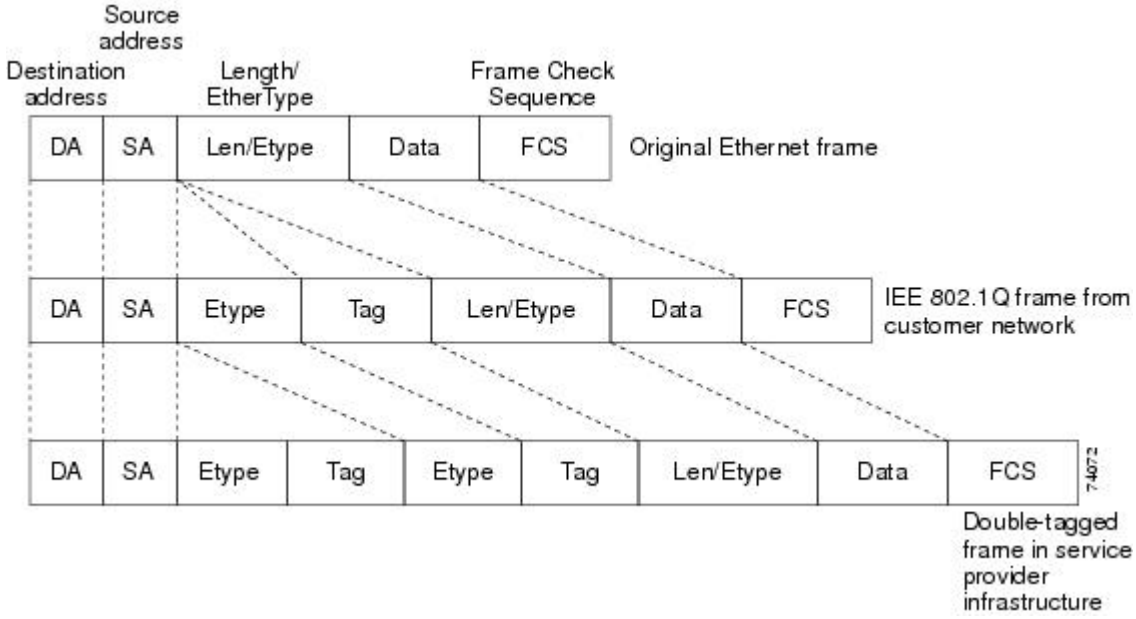
Figure 2: MKA and MACsec Topology with Multiple SVLANs



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:

Procedure

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:

Procedure

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

Procedure

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

Procedure

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

Procedure

Step 1 **enable**

Step 2 **Example:**

```
show running-config | sec kc1
key chain kc1 macsec
 key 01
   cryptographic-algorithm aes-128-cmac
  key-string 12345678901234567890123456789012
 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