

Troubleshoot WAN MACSEC on Routers

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Topology](#)

[MACSEC Overview To Troubleshoot](#)

[MACsec Packet Format](#)

[WAN-MACSEC](#)

[WAN MACSEC Packet Format](#)

[WAN MACSEC Terminology](#)

[MACSEC Key Agreement Protocol \(MKA\) and Cryptography Overview](#)

[Pre-shared Keys](#)

[802.1x/EAP](#)

[Troubleshoot WAN MACSEC](#)

[Configuration](#)

[Operational Issues](#)

[Related Information](#)

Introduction

This document describes basic WAN MACSEC protocol to understand operation and troubleshoot for Cisco IOS[®] XE routers.

Prerequisites

Requirements

There are no specific prerequisites for this document.

Components Used

The information in this document is specific for Cisco IOS XE routers such as ASR 1000, ISR 4000 and Catalyst 8000 families. Look for for specific hardware and software MACSEC support.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Topology



Topology Diagram

MACSEC Overview To Troubleshoot

MACsec is an IEEE 802.1AE standard based Layer 2 hop-by-hop encryption that provides data confidentiality, data integrity and data origin authentication for media access independent protocols with AES-128 encryption, only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

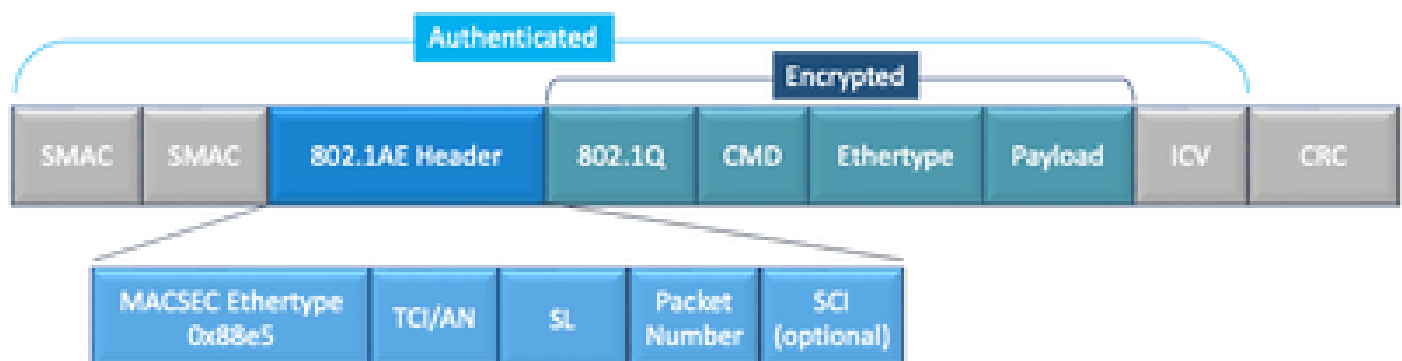
- Packets are decrypted on ingress port.
- Packets are clear in the device.
- Packets are encrypted on egress port.

MACsec provides secure communication on wired LANs, when MACsec is used to secure the communication between endpoints on a LAN, each packet on the wire is encrypted using symmetric key cryptography, so that communication can not be monitored or altered on the wire. When MACsec is used in conjunction with security group tags (SGTs), it provides protection for the tag along with the data contained in the payload of the frame.

MACsec provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying.

MACsec Packet Format

With 802.1AE (MACsec), frames are encrypted and protected with an integrity check value (ICV) with no impact to IP MTU or fragmentation and minimum L2 MTU impact: ~40 bytes (less than baby giant frame).



- **MACsec EtherType:** 0x88e5, designates that frame is a MACsec frame.
- **TCI/AN:** TAG Control Information/Association Number. Is the MACsec version number if confidentiality or integrity are used alone.
- **SL:** Length of the encrypted data.
- **PN:** Packet number used for replay protection.
- **SCI:** Secure Channel Identifier. Each connectivity association (CA) is a virtual port (MAC address of the physical interface plus 16-bit port ID).
- **ICV:** Integrity Check Value.

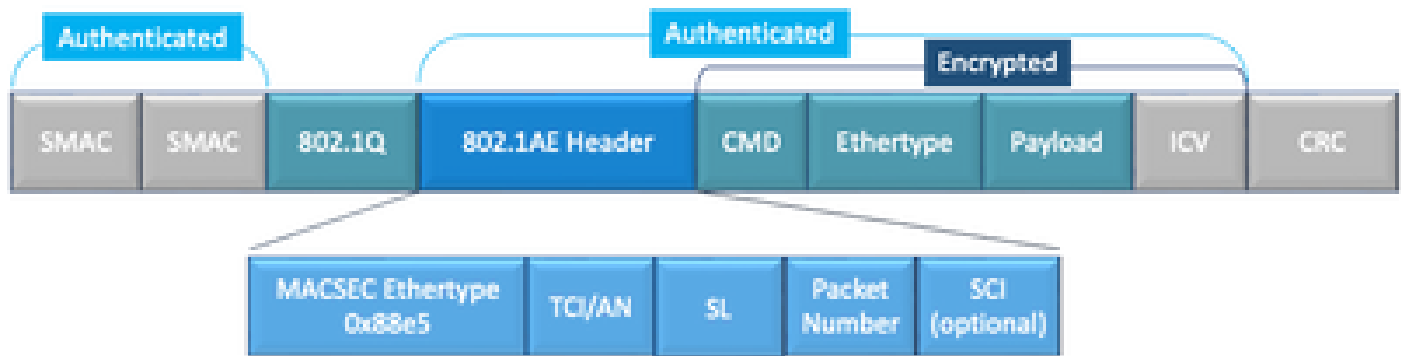
WAN-MACSEC

Ethernet has evolved beyond a private LAN transport, to include a variety of WAN or MAN transport options. WAN MACSEC provides end-to-end encryption across Layer 2 Ethernet WAN service either point-to-point or point-to-multipoint using AES 128 or 256-bit.

WAN MACsec is based on (LAN) MACsec, hence the name (and separate from IPsec), but offers several additional capabilities not available earlier.

WAN MACSEC Packet Format

There is a possibility that service provider does not support MACsec ethertype and can not differentiate L2 service if tag is encrypted so WAN MACSEC encrypts all of the frame after 802.1Q headers:



WAN MACSEC 802.1Q Tag in the Clear Packet Format Example

One of the new enhancements includes 802.1Q Tags in the Clear (aka ClearTag). This enhancement enables the ability to expose the 802.1Q tag outside of the encrypted MACsec header. Exposing this field provides several design options with MACsec, and in for public Carrier Ethernet transport providers, it is necessary for leveraging certain transport services.

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.

WAN MACSEC Terminology

MKA	MACSec Key Agreement, defined in IEEE 802.1XREV-2010 - Key agreement Protocol for discovering MACSec peers and negotiating keys.
-----	--

MSK	Master Session Key, generated during EAP exchange. Supplicant and authentication server use the MSK to generate CAK
CAK	Connectivity Association Key is derived from MSK. Is a long-lived master key used to generate all other keys used for MACSec.
CKN	Connectivity Association Key Name - identifies the CAK.
SAK	Secure Association Key - Derived from the CAK and is the key used by supplicant and switch to encrypt traffic for a given session.
KS	Key Server responsible for: <ul style="list-style-type: none"> • Selecting and advertising a cipher suite • Generating the SAK from the CAK.
KEK	Key Encrypting Key - used to protect MACsec keys (SAK)

MACSEC Key Agreement Protocol (MKA) and Cryptography Overview

MKA is the control plane mechanism used by WAN MACsec; specified in IEEE Std 802.1X which discovers mutually authenticated MACsec peers plus the next actions:

- Establishes and manages a CA (Connectivity Association).
- Manages Live/Potential peer list.
- Cipher suite negotiation.
- Elects Key Server (KS) among the members of a CA.
- Secure Association Key (SAK) derivation and management.
- Secure Key distribution.
- Key installation.
- Rekey.

One member gets elected as the Key server based on configured key-server priority (lowest), if the KS priority is same among peers, then, the lowest SCI wins.

KS generates a SAK only after all potential peers have become live and there is, at least, one live peer. It distributes the SAK and the cipher used to other participants using the MKA PDU or MKPDU in an encrypted format.

Participants check the cipher sent by the SAK and install it if it is supported, using it on every MKPDU to indicate the latest key they have; else, they shall reject SAK

When no MKPDU is received from a participants after 3 heartbeat (each heartbeat is of 2 seconds by default), peers are deleted from the live peer list; for instance, 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.

For this process, there are two methods to drive encryption keys:

- Pre-shared Keys
- 802.1x/EAP

Pre-shared Keys

If you use pre-shared keys, CAK=PSK and CKN must be manually entered. For key life time, ensure you have a key rollover and overlap during re-key time to:

- Exchange and install new SAK key and bind it to idle SA.
- Purge the old SAK key and allocate a new idle SA.

Configuration example:

```
<#root>
key chain
M_Key
  macsec

key 01
  cryptographic-algorithm
    aes-128-cmac
  key-string
    12345678901234567890123456789001
  lifetime 12:59:59 Oct 1 2023 duration 5000
key 02
  cryptographic-algorithm aes-128-cmac
  key-string 12345678901234567890123456789002
  lifetime 14:00:00 Oct 1 2023 16:15:00 Oct 1 2023
key 03
  cryptographic-algorithm aes-128-cmac
  key-string 12345678901234567890123456789003
  lifetime 16:15:00 Oct 1 2023 17:15:00 Oct 1 2023
key 04
  cryptographic-algorithm aes-128-cmac
  key-string 12345678901234567890123456789012
  lifetime 17:00:00 Oct 1 2023 infinite
```

Where bold words refers to:

M_Key: Key chain name.

key 01: Connectivity Association Key Name (same as CKN).


aes-128-cmac: MKA Authentication Cipher.

12345678901234567890123456789012: Connectivity Association Key (CAK).

Define policy:


```
<#root>
mka policy example
  macsec-cipher-suite
```

Where gcm-aes-256 refers to cipher suite(s) for secure association key (SAK) derivation.

 **Note:** This is basic policy configuration, more options such as **confidentiality-offset**, **sak-rekey**, **include-icv-indicator** and more are available for use depends on implementation.

Interface:

```
interface TenGigabitEthernet0/1/2
  mtu 2000
  ip address 198.51.100.1 255.255.255.0
  ip mtu 1468
  eapol destination-address broadcast-address
  mka policy example
  mka pre-shared-key key-chain M_Key
  macsec
end
```

 **Note:** If no mka policy is configured or applied, default policy is enabled and can be reviewed via **show mka default-policy detail**.

802.1x/EAP

If you use EAP method, all keys are generated from the Master Session Key (MSK). With IEEE 802.1X Extensible Authentication Protocol (EAP) framework, MKA exchanges EAPoL-MKA frames between devices, the Ether Type of EAPoL frames are 0x888E while the packet body in an EAPoL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). Those EAPoL frames contain the CKN of the sender, key server priority, and the MACsec capabilities.

 **Note:** By default, the switches process EAPoL-MKA frames but does not forward them.

Certificate-based MACsec Encryption configuration example:

Enrolling the Certificate (requires Certificate Authority):

```
crypto pki trustpoint EXAMPLE-CA
  enrollment terminal
  subject-name CN=ASR1000@user.example, C=IN, ST=KA, OU=ENG,O=Example
  revocation-check none
  rsakeypair mkaioscarsa
  storage nvram:
```

```
crypto pki authenticate EXAMPLE-CA
```

802.1x Authentication and AAA Configuration needed:

```
aaa new-model
dot1x system-auth-control
radius server ISE
  address ipv4 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
```

EAP-TLS Profile and 802.1X Credentials:

```
eap profile EAPTLS-PROF-IOSCA
  method tls
  pki-trustpoint EXAMPLE-CA
!
dot1x credentials EAPTLSCRED-IOSCA
  username asr1000@user.example
  pki-trustpoint EXAMPLE-CA
!
```

Interface:

```
interface TenGigabitEthernet0/1/2
  macsec network-link
  authentication periodic
  authentication timer reauthenticate
  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
```

Troubleshoot WAN MACSEC

Configuration

Check proper configuration and implementation support depending on platform; keys and parameters must match. Some of the common logs to identify if there is a problem on configuration are the next ones:

%MKA-3-INVALID_MACSEC_CAPABILITY : Terminating MKA Session because no peers had the required MACsec Cap

Check the MACsec capability of the peers' hardware or lower the requirements for MACsec capability by changing the MACsec configuration for the interface.

%MKA-3-INVALID_PARAM_SET : %s, Local-TxSCI %s, Peer-RxSCI %s, Audit-SessionID %s

There are some optional parameters that router can expect or not based on configuration and different default settings of the platform, ensure you include or discard on configuration.

%MKA-4-MKA_MACSEC_CIPHER_MISMATCH: Lower/Higher strength MKA-cipher than macsec-cipher for RxSCI %s, Au

There is a configuration mismatch on policy cipher suite, ensure proper match.

%MKA-3-MKPDU_VALIDATE_FAILURE : MKPDU validation failed for Local-TxSCI %s, Peer-RxSCI %s, Audit-Session

MKPDU failed one or more of the next validation checks:

- Valid MAC Address and EAPOL Header: Check both interfaces configuration, packet capture on ingress interface can corroborate current values.
- Valid CKN and Algorithm Agility: Ensure valid keys and algorithm suites.
- ICV verification: ICV verification is an optional parameter, configuration both ends must match.
- Correct order existence of MKA payloads: Possible interoperability issue.
- MI verification if peers exist: Member Identifier verification, unique for each participant.
- MN verification if peers exist: Message number verification, unique on every MKPDU transmitted and increments on every transmission.

Operational Issues

Once configuration is set, you can see %MKA-5-SESSION_START message but need to check if session comes up, a good command to start with is **show mka sessions [interface interface_name]**:

```
<#root>
```

```
Router1#
```

```
show mka sessions
```

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


Interface Port-ID	Local-TxSCI Peer-RxSCI	Policy-Name MACsec-Peers	Inherited Status	Key-Server CKN
Te0/1/2	40b5.c133.0e8a/0012			

Example

NO

NO

18 40b5.c133.020a/0012 1

Secured

01

Status refers to control plane session; Secured means Rx and Tx SAK is installed, if not, then it shows up as Not Secured.

- If status stays on Init, check physical interface state, connectivity via ping for peers and configuration match. At this point there is no MKPDU received and live peers, some platforms do padding while some other do not; consider up to 32 bytes of header overhead and ensure larger MTU for proper operation.
- If status stays on Pending, check if MKPDU are dropped either ingress or egress in control plane or interfaces errors/drops.
- If status stays on Not Secured, MKA interface is up and MKPDUs are flowing through but SAK is not installed, in this case the next log is seen:

```
%MKA-5-SESSION_UNSECURED : MKA Session was not secured for Local-TxSCI %s, Peer-RxSCI %s, Audit-Session
```

This is due to no MACsec support, invalid MACsec configuration, or other MKA failure on local or peer side prior to the establishment of a Secure Channel (SC) and installation of Secure Associations (SA) in MACsec. You can use detail command for further information **show mka session [interface interface_name] detail**:

```
<#root>
```

```
Router1#
```

```
show mka sessions detail
```

```
MKA Detailed Status for MKA Session
```

```
=====  
Status: SECURED - Secured MKA Session with MACsec
```

Local Tx-SCI..... 40b5.c133.0e8a/0012
Interface MAC Address.... 40b5.c133.0e8a
MKA Port Identifier..... 18
Interface Name..... TenGigabitEthernet0/1/2
Audit Session ID.....

CAK Name (CKN)..... 01

Member Identifier (MI)... DC5F7E3E38F4210925AAC8CA
Message Number (MN)..... 14462
EAP Role..... NA
Key Server..... NO

MKA Cipher Suite..... AES-128-CMAC

Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... 272DA12A009CD0A3D313FADF00000001 (1)
Old SAK Status..... FIRST-SAK
Old SAK AN..... 0
Old SAK KI (KN)..... FIRST-SAK (0)

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

MKA Policy Name..... Example
Key Server Priority..... 2
Delay Protection..... NO
Delay Protection Timer..... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility..... 80C201
SAK Rekey On Live Peer Loss..... NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation... NO
SAK Cipher Suite..... 0080C20001000002 (GCM-AES-256)
MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired..... YES

of MACsec Capable Live Peers..... 1
of MACsec Capable Live Peers Responded.. 0

Live Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority	RxSA Installed	SSCI
272DA12A009CD0A3D313FADF	14712	40b5.c133.020a/0012	1	YES	0

Potential Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority	RxSA Installed	SSCI
----	----	---------------	----------------	-------------------	------

Look for SAK information on peers and relevant data highlighted to better understand situation, if different SAK is in place, examine key used and lifetime or SAK rekey options configured, if pre-shared keys are used you can use **show mka keychains**:

```
<#root>
```

```
Router1#
```

```
show mka keychains
```

```
MKA PSK Keychain(s) Summary...
```

Keychain Name	Latest CKN Latest CAK	Interface(s) Applied
---------------	--------------------------	-------------------------

```
=====
```

```
Master_Key
```

```
01
```

```
<HIDDEN>
```

```
Te0/1/2
```

CAK is never shown but you can corroborate keychain name and CKN.

If session has been established but you have flaps or intermittent traffic flow, you must check if MKPDUs are flowing correctly among peers, if there is a timeout, you can see the next message:

```
%MKA-4-KEEPALIVE_TIMEOUT : Keepalive Timeout for Local-TxSCI %s, Peer-RxSCI %s, Audit-SessionID %s, CKN
```

If there is one peer, MKA Session is terminated, in case you have multiple peers and MKA has not received a MKPDU from one of its peers for more than 6 seconds, Live Peer is removed from the Live Peers List, you can start with **show mka statistics [interface interface_name]**:

```
<#root>
```

```
Router1#
```

```
show mka statistics interface TenGigabitEthernet0/1/2
```

```
MKA Statistics for Session
```

```
=====
```

```
Reauthentication Attempts.. 0
```

```
CA Statistics
```

```
Pairwise CAKs Derived... 0
```

```
Pairwise CAK Rekeys..... 0
```

```
Group CAKs Generated.... 0
```

```
Group CAKs Received..... 0
```

```
SA Statistics
```

```
SAKs Generated..... 0
```

```
SAKs Rekeyed..... 0
```

```
SAKs Received..... 1
```

```
SAK Responses Received.. 0
```

```
MKPDU Statistics
```

```
MKPDUs Validated & Rx... 11647
```

```
"Distributed SAK".. 1  
"Distributed CAK".. 0
```

```
MKPDUs Transmitted..... 11648
```

```
"Distributed SAK".. 0  
"Distributed CAK".. 0
```


MKPDUs transmitted and received must have similar numbers for one peer, ensure they increase at Rx and Tx both ends, to determine or guide the problematic direction, if there are differences you can enable **debug mka linksec-interface frames** both ends:

```
*Sep 20 21:14:10.803: MKA-LLI-MKPDU: Received CKN length (2 bytes) from Peer with CKN 01  
*Sep 20 21:14:10.803: MKA-LLI-MKPDU: MKPDU Received: Interface: [Te0/1/2 : 18] Peer MAC: 40:B5:C1:33:02  
*Sep 20 21:14:12.101: MKA-LLI-MKPDU: MKPDU transmitted: Interface [Te0/1/2: 18] with CKN 01  
*Sep 20 21:14:12.803: MKA-LLI-MKPDU: Received CKN length (2 bytes) from Peer with CKN 01  
*Sep 20 21:14:12.803: MKA-LLI-MKPDU: MKPDU Received: Interface: [Te0/1/2 : 18] Peer MAC: 40:B5:C1:33:02
```

In case there are no MKPDU received, look for incoming interface errors or drops, status of the peers interfaces and mka session; in case you have both routers sending but not receiving, MKPDUs are lost on the media and need to check intermediate devices for correct forwarding.

If you are not sending MKPDUs, check for physical interface state (line and errors/drops) and configuration; examine if you are generating those packets at control plane level, FIA trace and Embedded Packet Capture (EPC) are reliable tools for this purpose. Refer to [Troubleshoot with the Cisco IOS XE Datapath Packet Trace Feature](#)

You can use **debug mka events** and look for reasons can guide next steps.

 **Note:** Please use with caution **debug mka** and **debug mka diagnostics** as they show state machine and very detailed information that can cause control plane issues on the router.

If session is secured and stable but traffic is not flowing, check for encrypted traffic sending both peers:

```
<#root>
```

```
Router1#
```

```
show macsec statistics interface TenGigabitEthernet 0/1/2
```

```
MACsec Statistics for TenGigabitEthernet0/1/2
```

```
SecY Counters
```

```
Ingress Untag Pkts:      0  
Ingress No Tag Pkts:    0  
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: 98020

Egress Untag Pkts: 0
Egress Too Long Pkts: 0
Egress Protected Octets: 0

Egress Encrypted Octets: 98012

Controlled Port Counters

IF In Octets: 595380
IF In Packets: 5245
IF In Discard: 0
IF In Errors: 0
IF Out Octets: 596080
IF Out Packets: 5254
IF Out Errors: 0

Transmit SC Counters (SCI: 40B5C1330E8B0013)

Out Pkts Protected: 0

Out Pkts Encrypted: 970

Transmit SA Counters (AN 0)

Out Pkts Protected: 0

Out Pkts Encrypted: 970

Receive SA Counters (SCI: 40B5C133020B0013 AN 0)

In Pkts Unchecked: 0
In Pkts Delayed: 0

In Pkts OK: 967

In Pkts Invalid: 0

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

SecY Counters are current packets on physical interface, while the others are related to the Tx Secure Channel means packets being encrypted and transmitted and Rx Secured Association means valid packets received on the interface.

More debugs such as **debug mka errors** and **debug mka packets** helps on identifying issues, please use this last one with precaution as can induce heavy logging.

Related Information

- [MACsec and MKA Configuration Guide](#)
- [Cisco Technical Support & Downloads](#)