

Configuring MACsec

This section describes how to configure MACsec on Cisco IR8340 Routers.

- MACsec Encryption Overview, on page 1
- Limitations and Restrictions, on page 1
- Media Access Control Security and MACsec Key Agreement, on page 2
- Configuring MACsec Encryption, on page 3

MACsec Encryption Overview

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Cisco IR8340 Router supports 802.1AE encryption with MACsec Key Agreement (MKA) on switch-to-host links for encryption between the switch and host device. IR8340 also supports MACsec encryption for switch-to-switch (inter-network device) security using MKA-based key exchange protocol.

Note HSEC license is required to configure MACsec encryption.

Table 1: MACsec Support on Switch Ports

Connections	MACsec support
Switch-to-host	MACsec MKA encryption
Switch-to-switch	MACsec MKA encryption

MKA is supported on switch-to-host facing links. Host-facing links typically use flexible authentication ordering for handling heterogeneous devices with or without IEEE 802.1x, and can optionally use MKA-based MACsec encryption.

Limitations and Restrictions

- MACsec configuration is not supported on EtherChannel ports. Instead, MACsec configuration can be applied/removed on the individual member ports of an EtherChannel. To add/remove MACsec configuration, you must first unbundle the member ports from the EtherChannel.

- Configure either MACsec or PRP/HSR on the ports.
- Until PCH is available, MACsec and PTP are mutually exclusive.
- Packet number based rekey is not supported.
- Certificate based MKA (switch to switch) is not supported.
- VLAN Tag-in-clear is not supported.
- Legacy Cisco SAP (switch to switch) is not supported.
- Extended Packet Numbering (XPN) is not supported.

Media Access Control Security and MACsec Key Agreement

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

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

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

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

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



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

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

Configuring MACsec Encryption

Configuring MKA and MACsec

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

Configuring an MKA Policy

Follow these steps to configure an MKA policy.



Note After changing any MKA policy or MACsec configuration for active sessions, execute the **shutdown** command, and then the **no shutdown** command on a port, so that the changes are applied to active sessions.

Procedure

	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`	Enter global configuration mode.
Step 3	**mka policy** *policy name* **Example:** `Router(config)# mka policy mka_policy`	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 the MKA policy will always be GCM-AES-128.
Step 4	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**} **Example:** `Router(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	Configures a cipher suite for deriving SAK with 128-bit or 256-bit encryption.
Step 5	**end** **Example:** `Router(config-mka-policy)# end`	Exit enters MKA policy configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show mka policy** **Example:** `Router# show mka policy`	Displays MKA policy configuration information.

Configuring MACsec MKA using PSK

Procedure

	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`	Enter global configuration mode.
Step 3	**key chain** *key-chain-name* **macsec** **Example:** `Router(config)# key chain keychain1 macsec`	Configures a key chain and enters the key chain configuration mode.
Step 4	**key** *hex string* **Example:** `Router(config-key-chain)# key 1000`	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. **Note** For 128-bit encryption, use any value between 1 and 32 hex digit key-string. For 256-bit encryption, use 64 hex digit key-string.
Step 5	**cryptographic-algorithm** {**aes-128-cmac** | **aes-256-cmac**} **Example:** `Device(config-key-chain)# cryptographic-algorithm aes-128-cmac`	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 6	**key-string** { [**0**|**6**|**7**] *pwd-string* | *pwd-string*} **Example:** `Device(config-key-chain)# key-string 12345678901234567890123456789012`	Sets the password for a key string. Only hex characters must be entered.
Step 7	**end** **Example:** `Router(config-key-chain)# end`	Exits key chain configuration mode and returns to privileged EXEC mode.

Configuring MACsec MKA on an Interface using PSK

Procedure

	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`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface GigabitEthernet 0/1/4`	Enters interface configuration mode.
Step 4	**switchport mode access** **Example:** `Router(config-if)# switchport mode access`	Sets the switchport mode to access.
Step 5	**switchport access vlan** *vlan-id* **Example:** `Router(config-if)# switchport access vlan 203`	Specifies the VLAN for which this access port will carry traffic.
Step 6	**macsec network-link** **Example:** `Router(config-if)# macsec network-link`	Enables MACsec on the interface.
Step 7	**mka policy** *policy name* **Example:** `Router(config-if)# mka policy MKA_128`	Configures an MKA policy.
Step 8	**mka pre-shared-key key-chain** *key-chain-name* **Example:** `Router(config-if)# mka pre-shared-key key-chain KEY128`	Configures an MKA pre-shared-key key-chain name. **Note** The MKA pre-shared key can be configured on either physical interface or sub-interfaces and not on both.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Sample Configuration of Switch-to-Switch MACsec

This section shows sample configuration of switch-to-switch MACsec.

Device 1 Configuration

```
configure terminal
mka policy MKA_128
macsec-cipher-suite gcm-aes-128
key chain KEY128 macsec
key 1111
cryptographic-algorithm aes-128-cmac
key-string 11111111111111111111111111111111
end

configure terminal
interface Vlan203
ip address 22.1.1.1 255.255.255.0
end

configure terminal
interface GigabitEthernet0/1/4
switchport mode access
switchport access vlan 203
mka policy MKA_128
mka pre-shared-key key-chain KEY128
macsec network-link
no shutdown
end
```

Device 2 Configuration

```
configure terminal
mka policy MKA_128
macsec-cipher-suite gcm-aes-128
key chain KEY128 macsec
key 1111
cryptographic-algorithm aes-128-cmac
key-string 11111111111111111111111111111111
end

configure terminal
interface Vlan203
ip address 22.1.1.2 255.255.255.0
end

configure terminal
interface GigabitEthernet0/1/4
switchport mode access
switchport access vlan 203
mka policy MKA_128
mka pre-shared-key key-chain KEY128
macsec network-link
no shutdown
end
```

Example: Sample Configuration of Switch-to-Host MACsec

This section shows sample configuration of switch-to-host MACsec.

```
aaa new-model
!
!
```

```
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting network default start-stop group radius
aaa accounting system default start-stop group radius
!
!
aaa server radius dynamic-author
client <radius-server ip> server-key <key>
port 3799
!
aaa session-id common
!
!
ip dhcp-server 17.0.0.1
!
ip dhcp pool VLAN17
network 17.0.0.0 255.255.255.0
default-router 17.0.0.1
!
!
device-tracking policy DEVICE_MACSEC
no protocol udp
tracking enable
!
authentication critical recovery delay 300
!
service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE
linksec policy must-secure
!
dot1x system-auth-control
!
class-map type control subscriber match-all DOT1X
match method dot1x
!
class-map type control subscriber match-all DOT1X_FAILED
match method dot1x
match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_MEDIUM_PRIO
match authorizing-method-priority gt 20
!
class-map type control subscriber match-all DOT1X_NO_RESP
match method dot1x
match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all DOT1X_TIMEOUT
match method dot1x
match result-type method dot1x method-timeout
!
class-map type control subscriber match-all LINKSEC_FAIL_DOT1X
match authorization-fail linksec-failed
match method dot1x
!
class-map type control subscriber match-all LINKSEC_FAIL_MAB
match authorization-fail linksec-failed
match method mab
!
class-map type control subscriber match-all MAB_FAILED
match method mab
match result-type method mab authoritative
!

policy-map type control subscriber POLICY_MUSTSECURE
```

```
event session-started match-all
10 class always do-until-failure
10 authenticate using dot1x priority 10
event authentication-failure match-first
5 class DOT1X_FAILED do-until-failure
10 terminate dot1x
20 authenticate using mab priority 20
10 class DOT1X_NO_RESP do-until-failure
10 terminate dot1x
20 authenticate using mab priority 20
20 class MAB_FAILED do-until-failure
10 terminate mab
20 authentication-restart 60
40 class always do-until-failure
10 terminate dot1x
20 terminate mab
30 authentication-restart 60
event agent-found match-all
10 class always do-until-failure
10 terminate mab
20 authenticate using dot1x priority 10
event authentication-success match-all
10 class always do-until-failure
10 activate service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE
!
interface GigabitEthernet0/1/1
switchport access vlan 17
switchport mode access
device-tracking attach-policy DEVICE_MACSEC
macsec
access-session host-mode multi-host
access-session closed
access-session port-control auto
dot1x pae authenticator
service-policy type control subscriber POLICY_SHOULDSECURE
!
!
radius-server attribute 8 include-in-access-req
radius-server dead-criteria time 20 tries 2
radius-server timeout 20
radius-server deadtime 5
radius-server key <radius key>

radius server ACS
address ipv4 <radius-server ip> auth-port 1812 acct-port 1813
!
```

Verifying the Configuration

Use the following command to display MKA sessions on the interface:

```
# show mka sessions interface GigabitEthernet0/1/4

Summary of All Currently Active MKA Sessions on Interface GigabitEthernet0/1/4...

===========================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
===========================================================================================
Gi0/1/4        b08b.d071.86ac/000e POLICY           NO                YES
14             b08b.d071.86ac/0000 0                Init              1000
```

Use the following command to display MACsec status on the interface:

```
# show macsec status interface GigabitEthernet0/1/5
 MACsec is enabled
  Replay protect : enabled
  Replay window : 0
  Include SCI : yes
  Use ES Enable : no
  Use SCB Enable : no
  Admin Pt2Pt MAC : forceTrue(1)
  Pt2Pt MAC Operational : no
  Cipher : GCM-AES-128
  Confidentiality Offset : 0

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

Use the following command to display MACsec statistics on the interface:

```
# show macsec statistics interface GigabitEthernet0/1/5
 Transmit Secure Channels
  SCI : B08BD07186AD000F
  SC state : inUse(1)
   Elapsed time : 00:00:18
   Start time : 7w0d
   Current AN: 0
   Previous AN: -
   Next PN: 14
   SA State: inUse(1)
   Confidentiality : yes
   SAK Unchanged : yes
   SA Create time : 00:11:21
   SA Start time : 7w0d
   SC Statistics
    Auth-only Pkts : 0
    Auth-only Bytes : 0
    Encrypt Pkts : 0
    Encrypt Bytes : 0
   SA Statistics
    Auth-only Pkts : 0
    Encrypt Pkts : 13

  Port Statistics
   Egress untag pkts  0
   Egress long pkts  0

 Receive Secure Channels
  SCI : 0C75BDCC84A40007
  SC state : inUse(1)
   Elapsed time : 00:00:18
   Start time : 7w0d
   Current AN: 0
   Previous AN: -
   Next PN: 23
   RX SA Count: 0
   SA State: inUse(1)
```

```
  SAK Unchanged : yes
  SA Create time : 00:11:19
  SA Start time : 7w0d
  SC Statistics
   Notvalid pkts 0
   Invalid pkts 0
   Valid pkts 0
   Valid bytes 0
   Late pkts 0
   Uncheck pkts 0
   Delay pkts 0
   UnusedSA pkts 0
   NousingSA pkts 0
   Decrypt bytes 0
  SA Statistics
   Notvalid pkts 0
   Invalid pkts 0
   Valid pkts 21
   UnusedSA pkts 0
   NousingSA pkts 0

 Port Statistics
  Ingress untag pkts  0
  Ingress notag pkts  9
  Ingress badtag pkts  0
  Ingress unknownSCI pkts  0
  Ingress noSCI pkts  0
  Ingress overrun pkts  0
```

Use the following command to display detailed status for MKA session:

```
# show mka session interface GigabitEthernet0/1/5 detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. b08b.d071.86ad/000f
Interface MAC Address.... b08b.d071.86ad
MKA Port Identifier...... 15
Interface Name........... GigabitEthernet0/1/5
Audit Session ID.........
CAK Name (CKN)........... 1111
Member Identifier (MI)... CBA23DA1D3D77DF725CD43BB
Message Number (MN)...... 10
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)....... AD657F4EB0D237F5AFF9186F00000001 (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.......... Policy_128
Key Server Priority...... 0
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......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

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

Live Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------
  AD657F4EB0D237F5AFF9186F  8          0c75.bdcc.84a4/0007  0         YES            0

Potential Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------

Dormant Peers List:
  MI                       MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                            Priority  Installed
  ----------------------------------------------------------------------------------------
```