Configuring MACsec

About MACsec

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.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet. It offers the following capabilities:

- Provides line rate encryption.
- Ensures data confidentiality by providing strong encryption at Layer 2.
- Provides integrity checking to help ensure that data cannot be modified in transit.
- Key Lifetime and Hitless Key Rollover, on page 2
- Fallback Key, on page 2

Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and 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 keychain after the lifetime is expired. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Creating a MACsec Keychain, on page 9

A key can roll over to a second key within the same keychain by configuring the second key (in the keychain) 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, the key rolls over without traffic interruption).

Note The lifetime of the keys are overlapped to achieve hitless key rollover.

Fallback Key

A MACsec session can fail due to a key/key ID (CKN) mismatch or a finite key duration between the Fabric Interconnect and the peer. If a MACsec session fails, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

For more information, see Creating a MACsec Keychain.

Guidelines and Limitations for MACsec

MACsec functionality supports the following:

- Ethernet Uplink interfaces
- Ethernet Port-channel member link interfaces

- MKA is the only supported key exchange protocol for MACsec.

Note The Security Association Protocol (SAP) is not supported.

MACsec functionality does not support the following:

- Unified uplink
- FCoE uplinks
- Server, Storage, and Appliance ports
- QSA
- Link-level flow control (LLFC) and priority flow control (PFC)
- Multiple MACsec peers (different SCI values) for the same interface
- 1G port or any port on a MAC block that has 1G ports on it.

Note MACsec configuration is supported on end host mode only.

Cisco UCS Fabric Interconnect Limitations

Cisco UCS Manager 4.3(4a) release supports MACsec functionality from Cisco UCS 6454, Cisco UCS 64108, and Cisco UCS 6536 series fabric interconnects onwards.

Keychain Limitations

- You cannot overwrite the Key Hex String when the MACsec Keychain is applied on the interface. Instead, you must delete the old key and create the new key or a new keychain.
- For a given keychain, key activation time must overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.

Fallback Limitations

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and shows as rekeying on the old CA (Connectivity Association) under status. And the MACsec session on the new key on primary PSK will be in the Init state.
- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.
- The key ID (CKN) used in the fallback key chain must not match with any of the key IDs (CKNs) used in the primary key chain of the same switch interface and peer upstream switch interface.
- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec Policy Limitations

- BPDU packets can be transmitted before a MACsec session becomes secure.
- We recommend you to apply the same security policy **Should Secure-Should Secure** or **Must Secure-Must Secure** on the fabric interconnect and the peer switch interface.

Note Configuring MACsec with security-policy as **must-secure** on an Uplink Interface brings down the port, and the traffic drops until the MACsec session is secured.

Layer 2 Tunneling Protocol (L2TP) Restrictions

MACsec is not supported on ports that are configured for dot1q tunneling or L2TP.

MACsec EAPOL Limitations

- For enabling EAPOL (Extensible Authentication Protocol over LAN) configuration, the range of Ethernet type between 0 to 0x599 is invalid.
- While configuring EAPOL packets, the following combinations must not be used:
  - MAC Address 0100.0ccd.cdd0 with any ethertype
  - Any MAC Address with Ether types: 0xfff0, 0x800, 0x86dd
  - The default destination MAC address, 0180.c200.0003 with the default Ethernet type, 0x888e
  - Different EAPOL DMAC addresses and Ethertype on both MACsec peers. The MACsec session works only if the MACsec peer is sending MKAPDUs with the DMAC and Ethertype configured locally.
  - Within the same slice of the forwarding engine, EAPOL ethertype and dot1q ethertype cannot have the same value.
  - More than one custom EAPOL is not supported.
  - You cannot modify a custom EAPOL configuration if applied on any interface.

Statistics Limitations

- Statistics are cumulative.
- Few CRC errors may occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).
- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.

Enabling MACsec Configuration

Before you can access the MACsec commands, you must enable MACsec.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **enable**	Enable MACsec.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /macsec # **show**	Displays the MACsec configuration.

Example

The following example enables a MACsec configuration:

```
UCS-A# scope macsec
UCS-A /macsec# enable
UCS-A /macsec* # commit-buffer
UCS-A /macsec# show

MACsec Feature:
Admin State
-----------
Enabled
UCS-A /macsec
```

Disabling MACsec Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **disable**	Disables MACsec.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /macsec # **show**	Displays the MACsec configuration.

Example

The following example disables the MACsec encryption and commits the transaction:

```
UCS-A# scope macsec
UCS-A /macsec # disable
UCS-A /macsec* # commit-buffer
UCS-A /macsec# show

MACsec Feature:
Admin State
-----------
Disabled
UCS-A /macsec
```

Creating a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **create macsec-policy** *<name>*	Creates a MACsec policy.
Step 3	UCS-A /macsec/macsec-policy* # **set cipher-suite** { **gcm-aes-xpn-256** | **gcm-aes-xpn-128** | **gcm-aes-256** | **gcm-aes-128** }	Configure the cipher suite to be used for MACsec encryption. Configures one of the following ciphers: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256.
Step 4	UCS-A /macsec/macsec-policy* # **set key-server-priority** *<0-255>*	Enter the key server priority. You can enter a value between 0-255. Lower the value, higher the preference to be selected as the key server. Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.
Step 5	UCS-A /macsec/macsec-policy* # **set security-policy** { **should-secure** | **must-secure** }	Configures one of the following security policies to define the handling of data and control packets:

	Command or Action	Purpose
		• **must-secure**—Packets that do not carry MACsec headers are dropped. • **should-secure**—Packets that do not carry MACsec headers are permitted. This is the default value.
Step 6	UCS-A /macsec/macsec-policy* # **set replay-window-size** *<0-596000000>*	Configures the replay protection window such that the secured interface does not accept any packet that is less than the configured window size. The range is from 0 to 596000000.
Step 7	UCS-A /macsec/macsec-policy* # **set sak-expiry-time** *<60-2592000>*	Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0.
Step 8	UCS-A /macsec/macsec-policy* # **set confidentiality-offset** { **conf-offset-0** | **conf-offset-30** | **conf-offset-50** }	Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50.
Step 9	UCS-A /macsec/macsec-policy* # **set include-icv-indicator** { **yes** | **no** }	Configure the ICV for the frame arriving on the port.
Step 10	UCS-A /macsec/macsec-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # create macsec-policy macsec_policy
UCS-A /macsec/macsec-policy* # set cipher-suite gcm-aes-xpn-256
UCS-A /macsec/macsec-policy* # set key-server-priority 16
UCS-A /macsec/macsec-policy* # set security-policy should-secure
UCS-A /macsec/macsec-policy* # set replay-window-size 0
UCS-A /macsec/macsec-policy* # set sak-expiry-time 60
UCS-A /macsec/macsec-policy* # set confidentiality-offset conf-offset-0
UCS-A /macsec/macsec-policy* # set include-icv-indicator yes
UCS-A /macsec/macsec-policy* # commit-buffer
UCS-A /macsec/macsec-policy #
```

Viewing MACsec Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **show macsec-policy**	Displays the MACsec policy details.

Example

The following example shows how to view a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # show macsec-policy

MACsec Policy:
    MACsec Policy Name Cipher Suite    Key Server Priority Security Policy Repla
y Window Size SAK Expiry Time Confidentiality Offset Include ICV Indicator
    ------------------ --------------- ------------------- --------------- -----
------------ --------------- ---------------------- ---------------------
    default            GCM AES XPN 256 16                  Should Secure   14880
9600         0               Conf Offset 0          No
    test1              GCM AES XPN 256 16                  Should Secure   14880
9600         61              Conf Offset 0          No

UCS-A /macsec* #
```

Deleting a MACsec Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **delete macsec-policy** *<name>*	Deletes a MACsec policy.
Step 3	UCS-A /macsec # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # delete macsec-policy macsec_policy
```

```
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Creating a MACsec Keychain

- Only MACsec keychains result in converged MKA sessions.
- You can create a MACsec keychain and keys on the device.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **create macsec-keychain** *<name>*	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a MACsec Keychain, and commits the transaction:

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-keychain kc
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Viewing a MACsec Keychain

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **show macsec-keychain**	Displays the MACsec keychain details.

Example

The following example shows how to view a MACsec keychain:

```
UCS-A# scope macsec
UCS-A /macsec #  show macsec-keychain

Keychain:
    Keychain Name
    -------------
    test-kc-1
    test-kc-2
    test1

UCS-A /macsec #
```

Deleting a MACsec Keychain

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **delete macsec-keychain** *<name>*	Deletes the MACsec Keychain.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec keychain:

```
UCS-A# scope macsec
UCS-A /macsec # delete macsec-keychain kc
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Creating a MACsec Key

You can create a MACsec key on the device.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **create macsec-keychain** *<name>*	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.

	Command or Action	Purpose
Step 3	UCS-A /macsec/macsec-keychain* # **create macsec-key** *<id>*	Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64. **Note** The key must consist of an even number of characters.
Step 4	UCS-A /macsec/macsec-keychain* # **set key-hex-string** *<key>*	Key Hex String—Can contain from 32 to up to 144 hexadecimal characters. For a type-0 (un-encrypted key) the length of the key is 32 hexadecimal characters for an AES_128_CMAC cryptographic algorithm and 64 hexadecimal characters for an AES_256_CMAC cryptographic algorithm.
Step 5	UCS-A /macsec/macsec-keychain* # **set encrypt-type** { **type-0** | **type-7** }	The encrypt type includes the following: • Type 0—When the configured key-hex-string is an unencrypted string, type-0 must be selected. • Type 7—When the configured key-hex-string is a Type-7 encrypted string, this option must be selected.
Step 6	UCS-A /macsec/macsec-keychain* # **set cryptographic-algorithm** { **aes-128-cmac** | **aes-256-cmac** }	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 7	UCS-A /macsec/macsec-keychain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a MACsec key:

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-keychain kc
UCS-A /macsec/macsec-keychain* # create macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # set key
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789
UCS-A /macsec/macsec-keychain/macsec-key* # set encrypt-type type-0
UCS-A /macsec/macsec-keychain/macsec-key* # set cryptographic-algorithm aes-256-cmac
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Viewing MACsec Keys

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain* # **show macsec-key**	Displays the MACsec key configuration details.

Example

The following example shows how to view a MACsec key:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain* # show macsec-key

MACsec Key:
    Key ID     Key Hex String Encryption Type Cryptographic Algorithm
    ---------- -------------- --------------- ----------------------
    11         ****           Type 0          AES 256 CMAC
```

Deleting a MACsec Key

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **delete macsec-key** *<id>*	Deletes a MACsec Key.
Step 4	UCS-A /macsec/macsec-keychain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec Key:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # delete macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Creating a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec Keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec Key ID.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **create life-time**	Creates a MACsec Key Lifetime.
Step 5	UCS-A /macsec/macsec-keychain/macsec-key* # **set start-date-time***jan 1 2024 0 0 0*	The start-time argument is the time of day and date that the key becomes active.
Step 6	UCS-A /macsec/macsec-keychain/macsec-key* # **set end-date-time** *jan 2 2024 0 0 0*	The end-time argument is the time of day and date that the key becomes active.
Step 7	UCS-A /macsec/macsec-keychain/macsec-key* # **set duration** *<0-2147483646>*	The duration argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
Step 8	UCS-A /macsec/macsec-keychain/macsec-key* # **set timezone** { **local** | **UTC** }	The time zone of the key can be local or UTC. The default time zone is UTC.
Step 9	UCS-A /macsec/macsec-keychain/macsec-key* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain* # scope macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # create life-time
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set start-date-time jan 1 2024 0 0 0
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set end-date-time jan 2 2024 0 0 0
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set timezone local
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key/life-time #
```

Viewing a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec key configuration mode.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **show life-time**	Displays the Lifetime details.

Example

The following example shows how to view a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # scope macsec-key 11
UCS-A /macsec/macsec-keychain/macsec-key # show life-time

Life Time:
    Start Date Time         End Date Time           Timezone Duration(sec)
    ----------------------- ----------------------- -------- -------------
    2024-04-08T16:55:38.000 2024-04-08T16:55:38.000 Local    0
UCS-A /macsec/macsec-keychain/macsec-key #
```

Deleting a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec key configuration mode.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **delete life-time**	Deletes the Lifetime.
Step 5	UCS-A /macsec/macsec-keychain/macsec-key* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # scope macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key # delete life-time
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Creating a MACsec Interface Configuration

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **create macsec-interface-config** <*name*>	Create a MACsec interface configuration.
Step 3	UCS-A /macsec/macsec-interface-config* # **set key-chain-name** <*macsec-keychain-name*>	Sets the MACsec keychain name for the specified MACsec policy.
Step 4	UCS-A /macsec/macsec-interface-config* # **set policy-name** <*macsec-policy*>	Sets the MACsec policy name for the specified MACsec policy.
Step 5	UCS-A /macsec/macsec-interface-config* # **set fallback-keychain-name** <*macsec-keychain-name*>	Applies the MACsec configuration on a physical interface with a fallback keychain. It is optional to configure a fallback PSK. If a fallback keychain is configured, the fallback keychain along with the primary keychain ensures that the session remains active even if the primary keychain is mismatched, or there is no active key for the primary keychain.
Step 6	UCS-A /macsec/macsec-interface-config* # **set eapol-name** <*eapol-name*>	Applies the MACsec configuration on a physical interface with an EAPOL configuration. For more information on MACsec EAPOL, see Configurable EAPOL Destination and Ethernet Type.

	Command or Action	Purpose
Step 7	UCS-A /macsec/macsec-interface-config* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a MACsec interface configuration:

```
UCS-A scope macsec
UCS-A /macsec # create macsec-interface-config macsec_ifconfig
UCS-A /macsec/macsec-interface-config* # set key-chain-name kc
UCS-A /macsec/macsec-interface-config* # set policy-name macsec-policy
UCS-A /macsec/macsec-interface-config* # set fallback-keychain-name fb_kc
UCS-A /macsec/macsec-interface-config* # commit-buffer
UCS-A /macsec/macsec-interface-config #
```

Viewing MACsec Interface Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **show macsec-interface-config**	Displays the MACsec interface configuration details.

Example

The following example shows how to view a MACsec interface configuration:

```
UCS-A# scope macsec
UCS-A /macsec # show macsec-interface-config

Interface Configuration:
Interface Configuration Name Interface Keychain Name Interface Policy Name Fallback Keychain
 Name EAPOL Name
---------------------------- ---------------------- --------------------
---------------------- ----------
cus-eapol-m-t0 keychain-type0-aes128 mp-must fallback-type0-aes128 custom
cus-eapol-s-t7 keychain-type7-aes256 mp-should fallback-type7-aes256 custom
custom-eapol keychain-type0-aes256 mp-must fallback-type0-aes256 custom
dummy-config dummy-key default default
mic-m-t0-aes128 keychain-type0-aes128 mp-must fallback-type0-aes128 default
mic-m-t0-aes256 keychain-type0-aes256 mp-must fallback-type0-aes256 default
```

Deleting a MACsec Interface Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **delete macsec-interface-config** *<name>*	Deletes a MACsec interface configuration mode.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec interface configuration:

```
UCS-A scope macsec
UCS-A /macsec # delete macsec-interface-config macsec_ifconfig
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Configuring MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/fabric # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope interface** *<slot id>* *<port id>*	Specifies the interface that you are configuring.
Step 4	UCS-A# /eth-uplink/fabric/interface # **set macsec-intf-config-name** *<name>*	Sets the MACsec interface configuration name.
Step 5	UCS-A# /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink/fabric # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # set macsec-intf-config-name macsec_ifconfig
UCS-A# /eth-uplink/fabric/interface* # commit-buffer
UCS-A# /eth-uplink/fabric/interface #
```

Viewing MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric# **scope interface** <*name*>	Specifies the interface that you are configuring.
Step 4	UCS-A# /eth-uplink/fabric# **show interface** <*slot-id*> <*port-id*> **detail**	

Example

The following example show how to view MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # show interface detail
Interfaces:
    Slot Id: 1
    Port Id: 2
    User Label:
    Admin State: Enabled
    Oper State: Sfp Not Present
    State Reason: xcvr-absent
    flow control policy: default
    Speed: Auto
    Oper Speed: Auto
    Lic State: License Ok
    Grace Period: 0
    Ethernet Link Profile name: default
    Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
    Udld Oper State: Unknown
    MACsec Interface Config name: test-mic
    Licensing Message: Perpetual software license is installed. All ports on this Fabric
Interconnect are licensed
```

Deleting MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric# **scope interface** <*slot-id*> <*port-id*>	Enters the interface configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/interface # **set macsec-intf-config-name""**	Deletes the MACsec interface configuration name.
Step 5	UCS-A# /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink/fabric # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # set macsec-intf-config-name macsec_ifconfig
UCS-A# /eth-uplink/fabric/interface* # commit-buffer
UCS-A# /eth-uplink/fabric/interface #
```

Configuring MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **create port-channel** <*port-id*>	Creates a port channel.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **create member-port**<*slot-id*> <*port-id*>	Creates a member port channel.

	Command or Action	Purpose
Step 5	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **set macsec-intf-config-name** *<name>*	Sets the MACsec interface configuration name.
Step 6	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # create port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # create member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel/member-port* # set macsec-intf-config-name
macsec_ifconfig
UCS-A# /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A# /eth-uplink/fabric/port-channel/member-port #
```

Viewing MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope port-channel** *<port-id>*	Enters the port channel configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **scope member-port** *<slot-id>* *<port-id>*	Enters the member port configuration mode.
Step 5	UCS-A# /eth-uplink/fabric/port-channel* # **show detail**	Displays the uplink port channel member interface.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # scope member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel* # show detail
Member Ports:
Slot Id: 1
Port Id: 5
Membership: Down
```

```
Oper State: Sfp Not Present
State Reason: xcvr-absent
Lic State: License Ok
Grace Period: 0
Ethernet Link Profile name: default
Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
Udld Oper State: Unknown
MACsec Interface Config name: macsec_ifconfig
Licensing Message: Perpetual software license is installed. All ports on this Fabric
Interconnect are licensed
```

Deleting MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope port-channel** *<name>*	Enters the port channel configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **scope member-port** *<name>*	Enters the member port channel configuration mode.
Step 5	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **set macsec-intf-config-name ""**	Sets the MACsec interface configuration name.
Step 6	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # scope member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel/member-port* # set macsec-intf-config-name ""
UCS-A# /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A# /eth-uplink/fabric/port-channel/member-port #
```

Configurable EAPOL Destination and Ethernet Type

Configurable EAPOL MAC and Ethernet type provides you the ability to change the MAC address and the Ethernet type of the MKA packet, to allow CE device to form MKA sessions over the ethernet networks that consume the standard MKA packets.

The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

This feature is available at the interface level and the alternate EAPOL configuration can be changed on any interface at any given time as follows:

- If the MACsec is already configured on an interface, the sessions comes up with a new alternate EAPOL configuration.
- When MACsec is not configured on an interface, the EAPOL configuration is applied to the interface and is effective when MACsec is configured on that inferface.

Enabling EAPOL Configuration

You can enable the EAPOL configuration on any available interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec configuration mode.
Step 2	UCS-A /macsec # **create macsec-eapol**<*name*>	Creates a MACsec EAPOL configuration.
Step 3	UCS-A /macsec/macsec-eapol* # **set macaddress** *<AA:BB:CC:DD:EE:FF>*	Enables the MAC addresses.
Step 4	UCS-A /macsec/macsec-eapol* # **set ethertype** *<0x600-0xffff>*.	Enables the EAPOL configuration on the specified interface type and identity. If the ethernet type is not specified, the default ethernet type of MKA packets, which is 0x888e, is considered.
Step 5	UCS-A /macsec/macsec-eapol* # **exit**	Exits MACsec EAPOL configuration mode.
Step 6	UCS-A /macsec* # **scope macsec-interface-config** *<name>*.	Enters the MACsec interface configuration mode.
Step 7	UCS-A /macsec/macsec-interface-config* # **set eapol-name** *<eapol-name>*	Apply the MACsec EAPOL configuration on an interface.
Step 8	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 9	UCS-A /eth-uplink # **scope fabric** { **a** | **b** }	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).

	Command or Action	Purpose
Step 10	UCS-A /eth-uplink/fabric # **scope interface** <*slot-id*><*port-id*>	Dsiplays the Ethernet uplink fabric interconnect mode for the specified interface.
Step 11	UCS-A /eth-uplink/fabric/interface # **set macsec-interface-config-name** <*interface name*>	Sets the interface configuration name.
Step 12	UCS-A /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables a MACsec EAPOL configuration and applies it on an interface.

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-eapol custom-eapol
UCS-A /macsec/macsec-eapol* # set macaddress 65:25:22:22:15:71
UCS-A /macsec/macsec-eapol* # set ethertype 0x888e
UCS-A /macsec/macsec-eapol* # exit
UCS-A  /macsec* # scope macsec-interface-config <name>
UCS-A /macsec/macsec-interface-config* # set eapol-name <eapol-name>
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 1 4
UCS-A /eth-uplink/fabric/interface # set macsec-intf-config-name macsec-ifconfig
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/fabric/interface #
```

Disabling EAPOL Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **set interface** <*slot-id*> <*port-id*>	Sets the interface configuration name.
Step 4	UCS-A /eth-uplink/fabric/interface # **set macsec-intf-config-name** <*interface-name*>	Sets the MACsec interface configuration name.
Step 5	UCS-A /eth-uplink/fabric/interface/macsec-interface-config* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable a MACsec EAPOL configuration:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 1 4
UCS-A /eth-uplink/fabric/interface # set macsec-intf-config-name macsec-ifconfig
UCS-A /eth-uplink/fabric/interface* # commit-buffer
```

Displaying MACsec Sessions

The Operational states of the MACsec session on an interface are displayed as follows:

```
UCS-A /eth-uplink/fabric/interface # show macsec-session

Interface:

MACsec State            MACsec State Reason            MACsec Auth-Mode            MACsec
 Key-Server
----------------------- ------------------------------ ----------------------------
-----------------
Secured                  Secured MKA Session with MACsec Primary Psk                  No

Interface:

MACsec State              MACsec State Reason              MACsec Auth-Mode
MACsec Key-Server
------------------------ ------------------------------- ----------------------------
-----------------

UCS-A /eth-uplink/fabric/interface # show macsec-session detail
MACsec session:
    MACsec State: Secured
    MACsec State Reason: Secured MKA Session with MACsec
    MACsec Auth-Mode: Primary Psk
    MACsec Key-Server: No
    MACsec Cipher Suite: GCM AES XPN 256
    MACsec Confidentiality Offset: Conf Offset 0

    MACsec State:
    MACsec State Reason:
    MACsec Auth-Mode:
    MACsec Key-Server:
    MACsec Cipher Suite:
    MACsec Confidentiality Offset:
```

The possible values for operational states are as follows:

- MACsec Status—Init, Pending, Secured, Rekeyed
- MACsec Key-server—yes, no
- MACsec Auth-mode—Primary-PSK, Fallback-PSK

The following CLI will have two more additional possible values of **State Reason** to represent the state of interface based on status of the MACsec session configured on it.

```
UCS-A /eth-uplink/fabric/interface # show interface
```

```
Interface:

Slot Id    Port Id    Admin State Oper State       Lic State            Grace Period
State Reason Ethernet Link Profile name Oper Ethernet Link Profile name
---------- ---------- ----------- ---------------- -------------------- ---------------
------------ ------------------------- ------------------------------
1          1          Enabled     Link Down        License Ok           0
link-failure default                    fabric/lan/eth-link-prof-default
```

Displaying MACsec Statistics

You can display MACsec statistics using the following commands:

Command	Description
show stats macsec-tx-stats	Displays the MACsec transmitter status.
show stats macsec-rx-stats	Displays the MACsec receiver status.

The following example shows the MACsec security statistics for a specific Ethernet interface.

Note The following differences exist for uncontrolled and controlled packets in Rx and Tx statistics:

Rx statistics:

- Uncontrolled = Encrypted and unencrypted
- Controlled = Decrypted

Tx statistics:

- Uncontrolled = Unencrypted
- Controlled = Encrypted

The following example shows the MACsec statistics:

```
UCS-A /eth-uplink/fabric/interface # show stats ether-macsec-rx-stats

Ether Macsec Rx Stats:
Time Collected: 2024-05-07T15:59:30.243
Monitored Object: sys/switch-A/slot-1/switch-ether/port-8
Suspect: No
Unicast Uncontrolled Packets (packets): 459227
Multicast Uncontrolled Packets (packets): 3648755
Broadcast Uncontrolled Packets (packets): 9494097
Uncontrolled Rx Drop Packets (packets): 0
Uncontrolled Rx Error Packets (packets): 0
Unicast Controlled Packets (packets): 0
Multicast Controlled Packets (packets): 0
Broadcast Controlled Packets (packets): 0
Controlled Rx Drop Packets (packets): 0
Controlled Rx Error Packets (packets): 0
Controlled Packets: 12902005
Thresholded: Unicast Uncontrolled Packets Delta Min
```

```
UCS-A /eth-uplink/fabric/interface # show stats ether-macsec-tx-stats

Ether Macsec Tx Stats:
Time Collected: 2024-05-07T15:59:30.243
Monitored Object: sys/switch-A/slot-1/switch-ether/port-8
Suspect: No
Unicast Uncontrolled Packets (packets): 0
Multicast Uncontrolled Packets (packets): 0
Broadcast Uncontrolled Packets (packets): 0
Uncontrolled Rx Drop Packets (packets): 0
Uncontrolled Rx Error Packets (packets): 0
Unicast Controlled Packets (packets): 0
Multicast Controlled Packets (packets): 0
Broadcast Controlled Packets (packets): 0
Controlled Rx Drop Packets (packets): 0
Controlled Rx Error Packets (packets): 0
Controlled Packets: 883044
Thresholded: Unicast Uncontrolled Packets Delta Min
```