MACsec Commands

key chain

To create or modify a key chain, use the **key chain** command in the key chain configuration mode. To remove this configuration, use the **no** form of this command.

key chain *key-chain-name* **macsec**
no key chain *key-chain-name* **macsec**

Syntax Description

key-chain-name	Specifies the name of the keychain. The maximum length is 32 (128-bit encryption)/64 (256-bit encryption) character hexadecimal string.
macsec	Specifies the key chain for MACsec encryption.

Command Default

No default behavior or values.

Command Modes

Key chain configuration (config)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how you can configure a key chain for MACsec encryption:

```
Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)#
```

key

To create or modify a keychain key, use the **key** command in keychain-key configuration mode. To remove this configuration, use the **no** form of this command.

key *key-id*
no key *key-id*

Syntax Description

key-id	Hexadecimal string of 2 - 64 characters.

Command Default

No default behavior or values.

Command Modes

Key chain configuration (config)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

The key must be of even number of hex characters. Entering an odd number of characters will exit the MACsec configuration mode.

Examples

The following example shows how to use the **key** command:

```
Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
```

key-string

To specify the text string for the key, use the **key-string** command in key configuration submode under the macsec key chain mode.

To remove this configuration, use the **no** form of this command.

key-string [{ **clear** | **password** | **password6** }] *key-string-text* **cryptographic-algorithm** { **aes-128-cmac** | **aes-256-cmac** }
no key-string [{ **clear** | **password** | **password6** }] *key-string-text* **cryptographic-algorithm** { **aes-128-cmac** | **aes-256-cmac** }

Syntax Description

clear	Specifies the key string in clear-text form.
password	Specifies the key in encrypted form.
password6	Specifies the key in Type 6 encrypted form.
key-string-text	Text string for the key, which is encrypted by the parser process before being saved to the configuration. The text string has the following character limitations: • Plain-text key strings—Minimum of 1 character and a maximum of 32 (128-bit encryption)/64 (256-bit encryption) characters (hexadecimal string). • Encrypted key strings—Minimum of 4 characters and no maximum.

Command Default

The default value is clear.

Command Modes

Key configuration submode under the macsec key chain mode.

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

For an encrypted password to be valid, the following statements must be true:

- String must contain an even number of characters, with a minimum of four.
- The first two characters in the password string must be decimal numbers and the rest must be hexadecimals.
- The first two digits must not be a number greater than 53.

Either of the following examples would be valid encrypted passwords:

1234abcd or 50aefd

Examples

The following example shows how to use the **key-string** command:

For AES 128-bit encryption:

```
Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string 12345678123456781234567812345678
cryptographic-algorithm AES-128-CMAC
```

For AES 256-bit encryption with clear-text CAK:

```
Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string clear
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMACRP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#commit
```

cryptographic-algorithm

Configures the cryptographic algorithm used for authenticating a peer for MACsec encryption in the Keychain-key configuration mode.

To disable this feature, use the **no** form of this command.

cryptographic-algorithm *authentication algorithm*
no cryptographic-algorithm *authentication algorithm*

Syntax Description

authentication algorithm	Configures the 128-bit or 256-bit AES encryption algorithm.

Command Default

No default behavior or values.

Command Modes

Keychain-key configuration.

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

If you do not specify the cryptographic algorithm, MAC computation and API verification would be invalid.

Examples

The following example shows how to use the cryptographic-algorithm command for MACsec Encryption:

```
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string 11111111111111111111111111111111
cryptographic-algorithm aes-256-cmac


Examples

The following example shows how to use the AES-128-CMAC authentication algorithm command:
```

```
Device# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string 12345678123456781234567812345678
cryptographic-algorithm aes-128-cmac


Examples

The following example shows how to use the AES-256-CMAC authentication algorithm command:

Device# key chain mac_chain macsec
Device(config-mac_chain-MacSec) # key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string 12345678123456781234567812345678l
```

lifetime

Configures the validity period for the MACsec key or CKN in the Keychain-key configuration mode. To disable this feature, use the **no** form of this command.

The lifetime period can be configured with a duration in seconds, as a validity period between two dates (for example, Jan 01 2014 to Dec 31 2014), or with an infinite validity.

The key is valid from the time you configure in HH:MM:SS format. Duration is configured in seconds.

When a key has expired, the MACsec session is torn down and running the show macsec mka session command does not display any information. If you run the show macsec mka interface and show macsec mka interface detail commands, you can see that the session is unsecured.

cryptographic-algorithm *start_time start_date* { *end_time end_date* | **duration** *validity* | **infinite** }
no cryptographic-algorithm *start_time start_date* { *end_time end_date* | **duration** *validity* | **infinite** }

Syntax Description

start_time	Start time in hh:mm:ss from which the key becomes valid. The range is from 0:0:0 to 23:59:59.
end_time	End time in hh:mm:ss at which point the key becomes invalid. The range is from 0:0:0 to 23:59:59.
start_date	The date in DD month YYYY format that the key becomes valid.
end_date	The date in DD month YYYY format that the key becomes invalid.
duration*validity*	The key chain is valid for the duration you configure. You can configure duration in seconds.
infinite	The key chain is valid indefinitely.

Command Default

No default behavior or values.

Command Modes

Keychain-key configuration.

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to use the lifetime command:

```
! For AES 128-bit encryption

Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string 12345678123456781234567812345678
cryptographic-algorithm AES-128-CMAC
Device(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2015 12:00:00
30 september 2016

! For AES 256-bit encryption

Device(config)# key chain mac_chain macsec
Device(config-mac_chain-MacSec)# key 1234abcd5678
Device(config-mac_chain-MacSec-1234abcd5678)# key-string
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMAC
Device(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2015 12:00:00
30 september 2016
```

mka policy

To configure an MKA policy, use the **mka policy** command in Global Configuration mode. To disable this feature, use the **no** form of this command.

mka policy *policy-name*
no mka policy *policy-name*

Syntax Description

policy-name	Name of the MACsec policy for encryption.

Command Default

No default behavior or values.

Command Modes

Global Configuration mode

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to use the **macsec-policy** command:

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

confidentiality-offset

To enable MACsec Key Agreement protocol (MKA) to set the confidentiality offset for MACsec operations, use the **confidentiality-offset** command in MKA-policy configuration mode. To disable confidentiality offset, use the **no** form of this command.

confidentiality-offset
no confidentiality-offset

Command Default

Confidentiality offset is disabled.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to enable the **confidentiality offset** command:

```
Device(config)# mka policy mka-policy
Device(config-mka-policy)# confidentiality-offset
```

delay-protection

To configure MKA to use delay protection in sending MACsec Key Agreement Protocol Data Units (MKPDUs), use the **delay-protection** command in MKA-policy configuration mode. To disable delay protection, use the **no** form of this command.

delay-protection
no delay-protection

Command Default

Delay protection for sending MKPDUs is disabled.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to configure MKA to use delay protection in sending MKPDUs:

```
Device(config-mka-policy)# delay-protection
```

include-icv-indicator

To include the integrity check value (ICV) indicator in MKPDU, use the **include-icv-indicator** command in MKA-policy configuration mode. To disable the ICV indicator, use the **no** form of this command.

include-icv-indicator
no include-icv-indicator

Command Default

ICV indicator is included.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to include the ICV indicator in MKPDU:

```
Device(config)# mka policy 2
Device(config-mka-policy)# include-icv-indicator
```

key-server

To configure MKA key-server options, use the **key-server** command in MKA-policy configuration mode. To disable MKA key-server options, use the **no** form of this command.

key-server priority *value*
no key-server priority *value*

Syntax Description

priority*value*	Specifies the priority value of the MKA key-server.

Command Default

MKA key-server is disabled.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to configure the MKA key-server:

```
Device(config)# mka policy 2
Device(config-mka-policy)# key-server priority 33
```

macsec-cipher-suite

To configure cipher suite for deriving Security Association Key (SAK), use the **macsec-cipher-suite** command in MKA-policy configuration mode. To disable cipher suite for SAK, use the **no** form of this command.

macsec-cipher-suite { **gcm-aes-128** | **gcm-aes-256** | **gcm-aes-xpn-128** | **gcm-aes-xpn-256** }

no macsec-cipher-suite { **gcm-aes-128** | **gcm-aes-256** | **gcm-aes-xpn-128** | **gcm-aes-xpn-256** }

Syntax Description

gcm-aes-128	Configures cipher suite for deriving SAK with 128-bit encryption.
gcm-aes-256	Configures cipher suite for deriving SAK with 256-bit encryption.
gcm-aes-xpn-128	Configures cipher suite for deriving SAK with 128-bit encryption for Extended Packet Numbering (XPN).
gcm-aes-xpn-256	Configures cipher suite for deriving SAK with 256-bit encryption for XPN.

Command Default

GCM-AES-128 encryption is enabled.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

If the device supports both GCM-AES-128 and GCM-AES-256 ciphers, it is highly recommended to define and use a user-defined MKA policy to include both or only 256 bits cipher, based on your requirements.

Examples

The following example shows how to configure MACsec cipher suite for deriving SAK with 256-bit encryption:

```
Device(config)# mka policy 2
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-256
```

sak-rekey

To configure the Security Association Key (SAK) rekey time interval for a defined MKA policy, use the **sak-rekey** command in MKA-policy configuration mode. To stop the SAK rekey timer, use the **no** form of this command.

sak-rekey { **interval** *time-interval* | **on-live-peer-loss** }

no sak-rekey { **interval** *time-interval* | **on-live-peer-loss** }

Syntax Description

interval*time-interval*	SAK rekey interval in seconds. The range is from 30 to 65535, and the default is 0.
on-live-peer-loss	Peer loss from the live membership.

Command Default

The SAK rekey timer is disabled. The default is 0.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to configure the SAK rekey interval:

```
Device(config)# mka policy 2
Device(config-mka-policy)# sak-rekey interval 300
```

use-updated-eth-header

To enable interoperability between devices and any port on a device that includes the updated Ethernet header in MACsec Key Agreement Protocol Data Units (MKPDUs) for integrity check value (ICV) calculation, use the **ssci-based-on-sci** command in MKA-policy configuration mode. To disable the updated ethernet header in MKPDUs for ICV calculation, use the **no** form of this command.

use-updated-eth-header
no use-updated-eth-header

Command Default

The Ethernet header for ICV calculation is disabled.

Command Modes

MKA-policy configuration (config-mka-policy)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

The updated Ethernet header is non-standard. Enabling this option ensures that an MACsec Key Agreement (MKA) session between the devices can be set up.

Examples

The following example shows how to use the **key** command:

```
Device(config)# mka policy 2
Device(config-mka-policy)# use-updated-eth-header
```

mka pre-shared-key

To configure MACsec Key Agreement (MKA) MACsec on a device interface using a Pre Shared Key (PSK), use the **mka pre-shared-key key-chain** command in interface configuration mode. To disable it, use the **no** form of this command.

mka pre-shared-key key-chain *key-chain-name*
no mka pre-shared-key key-chain *key-chain-name*

Syntax Description

mka pre-shared-key key-chain	Enables MACsec MKA configuration on device interfaces using a PSK.

Command Default

MKA pre-shared-key is disabled.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

This example shows how to configure MKA MACsec on an interface using a PSK:

```
Device(config)# interface Gigabitethernet 1/0/20
Device(config-if)# mka pre-shared-key key-chain kc1
```

fallback-key

To provide an alternative fallback option to maintain secure communications, use the **fallback-key** command in Interface configuration mode. Use the **fallback-key** command along with the **mka pre-shared-key key-chain** command. The **mka pre-shared-key key-chain** command is used to enable MKA with a pre-shared key for MACsec encryption on a specified interface.

To remove this configuration, use the **no** form of this command.

mka pre-shared-key key-chain *keychain-name* [{ **fallback-key-chain** *fallback-keychain-name* }]
no mka pre-shared-key key-chain *keychain-name* **fallback-key-chain** *fallback-keychain-name*

Syntax Description

keychain-name	Used as the primary key chain for MKA.
fallback-keychain-name	Used as the fallback key chain for MKA.

Command Default

No default behavior or values.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

The provided configuration enables MKA with a pre-shared key for MACsec encryption on the specified interface (TenGigabitEthernet 0/0/5).

MKA provides secure key agreement for MACsec, which is used to encrypt traffic on the interface.

The primary key-chain is used to store the primary pre-shared key .

MACsec's fallback key feature establishes an MKA session with the pre-shared fallback key whenever the pre-shared key fails to establish a session because of key mismatch.

Fallback key chain supports infinite lifetime with one key only. The connectivity association key name (CKN) ID used in the fallback key chain must not match any of the CKN IDs used in the primary key chain.

Examples

The following example shows how to use the **fallback-key** command:

```
Device(config-keychain)# interface TenGigabitEthernet 0/0/5
Devcie(config-if)# mka pre-shared-key key-chain mka-keychain128 fallback-key-chain
mka-keychain256
```

macsec access-control

To control the behavior of unencrypted packets, use the **macsec access-control** command in Interface configuration mode. To disable this option, use the **no** form of this command.

macsec access-control { **must-secure** | **should-secure** }
no macsec access-control { **must-secure** | **should-secure** }

Syntax Description

must-secure	Allows unencrypted packets from the physical interface or subinterfaces to be transmitted or received.
should-secure	Allow unencrypted packets from physical interface or subinterfaces to be transmitted or received. All such packets are dropped except for MKA control protocol packets.

Command Default

No default behavior or values.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

The **macsec access-control** command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

Examples

The following example shows how to use the **macsec access-control** command:

```
Device(config)#interface GigabitEthernet0/0/1
Device(config-if)# macsec access-control must-secure

Device(config)#interface GigabitEthernet0/0/1
Device(config-if)# macsec access-control should-secure
```

replay-protection window-size

To change the replay window size, use the **replay-protection window-size** command in Interface configuration mode. The range for window size is 0 to 4294967295. To turns off MACsec replay-protection, use the **no** form of this command.

replay-protection window-size *frames*
no replay-protection window-size

Syntax Description

frames	Enable replay protection, and configure the window size in number of frames. The range is from 0 to 4294967295. The default window size is 0.

Command Default

No default behavior or values.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

Replay protection is a feature provided by MACsec to counter replay attacks. Each encrypted packet is assigned a unique sequence number and the sequence is verified at the remote end. Frames transmitted through a Metro Ethernet service provider network are highly susceptible to reordering due to prioritization and load balancing mechanisms used within the network.

A replay window is necessary to support use of MACsec over provider networks that reorder frames. Frames within the window can be received out of order, but are not replay protected. The default window size is set to 64.

The replay protection window may be set to zero to enforce strict reception ordering and replay protection.

Note A replay protection window can be configured independently on either physical interface or subinterface. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on subinterface overrides the inherited value or policy for that sub-interface.

Examples

The following example shows how to use the **replay-protection window-size** command:

```
Device(config)#interface GigabitEthernet0/0/1
Device(config-if)# macsec replay-protection window-size 10
```

eapol

To configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface, use the **eapol** command in Interface configuration mode. To disble this option, use the **no** form of this command.

eapol *eth-type*
no eapol *eth-type*

Syntax Description

eth-type	Configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface.

Command Default

No default behavior or values.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Examples

The following example shows how to use the **eapol***eth-type* command:

```
Device(config)#interface GigabitEthernet0/0/1
Device(config-if)# eapol eth-type 0xB860
```

eapol destination-address

To change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider, use the **eapol destination-address** command in Interface configuration mode. To disable this option,, use the **no** form of this command.

eapol destination-address [{ *MAC-Address* | { **bridge-group-address** | **broadcast-address** | **lldp-multicast-address** } }]

no eapol destination-address [{ *MAC-Address* | { **bridge-group-address** | **broadcast-address** | **lldp-multicast-address** } }]

Syntax Description

MAC-Address	Configures an Extensible Authentication Protocol over LAN (EAPoL) destination MAC address on the interface.
bridge-group-address	Sets the destination address as a bridge group.
broadcast-address	Sets the destination address as a broadcast address.
lldp-multicast-address	Sets the destination address as a LLDP multicast address.

Command Default

No default behavior or values.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Command qualified for use in Cisco vManage templates.

Usage Guidelines

When the eapol destination-address command is configured on the main interface, it is applied to any subinterfaces on that interface. However, if the eapol destination-address command is configured on the subinterface, that takes take precedence over the command on the main interface.

Examples

The following example shows how to use the **eapol destination-address** command:

```
Device(config)#interface GigabitEthernet0/0/1
Device(config-if)# eapol destination-address 0018.b967.3cd0
Device(config-if)# eapol destination-address bridge-group-address
Device(config-if)# eapol destination-address broadcast-address
Device(config-if)# eapol destination-address lldp-multicast-address
```