WAN MACsec and MKA Support Enhancements

The WAN MACsec and MACsec Key Agreement protocol (MKA) features introduce MACsec support on WAN, and uplink support and pre-shared key support for the MKA.

Table 1: Feature History

Feature Name	Release	Description
MACsec support with PTP for 1GE NCS4200-1T16G-PS Interface Module	Cisco IOS XE Dublin 17.12.1	You can now configure MACsec support with Precision Time Protocol (PTP) packets for mitigating security vulnerabilities on a router.
MACsec Support with SyncE for 1GE and 10GE NCS4200-1T16G-PS Interface Module	Cisco IOS XE Dublin 17.10.1	You can now configure MACSec encryption on Synchronized Ethernet (SyncE) interfaces that send and receive Ethernet Synchronization Message Channel (ESMC) packets. The MACSec header is added to the ESMC packets to secure data on the physical media. Also, MACSec encryption prevents the higher-layer protocols' traffic from being compromised.
802.1AE WAN MACsec Enhancement for 1GE and 10GE NCS4200-1T16G-PS	Cisco IOS XE Cupertino 17.8.1	The 802.1AE WAN MACsec supports 10GE physical layer (PHY) interfaces for NCS4200-1T16G-PS interface module. From this release, full HA, Power on Self Test (POST) and double tag support are available on NCS4200-1T16G-PS interface module. The following new command is introduced: **show macsec post**
802.1AE WAN MACsec for 1GE and 10GE NCS4200-1T16G-PS	Cisco IOS XE Bengaluru 17.6.1	The WAN MACsec and MKA feature introduce MACsec support on WAN and uplink support and pre-shared key support for the MACsec Key Agreement protocol (MKA). The WAN MACsec supports 1GE and 10GE interfaces for NCS4200-1T16G-PS interface module.

Feature Name	Release	Description
MAC Security	Cisco IOS XE Bengaluru 17.5.1	The MACsec and MACsec Key Agreement protocol (MKA) features are introduced on the router interface (main interface) with pre-shared key support for the MKA. This feature is supported on the Cisco RSP3 module.

Prerequisites for WAN MACsec and MKA Support Enhancements

- Layer 2 transparent Ethernet Services must be present.
- The service provider network must provide a MACsec Layer 2 Control Protocol transparency such as, Extensible Authentication Protocol over LAN (EAPoL).

Restrictions for WAN MACsec and MKA Support Enhancements

- MACSec encryption of SyncE packets is not supported for those that are tagged with EFP service instances.
- MACsec with PTP is not supported until Cisco IOS XE Dublin 17.11.1.

  Starting with Cisco IOS XE Dublin 17.12.1, the router supports MACsec and PTP with G8275.1 profile only in different 1G interfaces on NCS4200-1T16G-PS Interface Module. You cannot configure different 1G on the same interface.
- For Ethernet service instance on the customer edge (CE) copper link to work, the Ethernet service instance, where xconnect is present in the provider edge (PE) must have the **remote link failure notification** command that is configured on the link.

  Following is a sample configuration.

```
interface TenGigabitEthernet x/y/z
  description ### X connect 1 <name> ###
  mtu 9216
  no ip address
  negotiation auto
  no keepalive
  service instance 10 ethernet
  encapsulation default
  l2protocol tunnel cdp stp vtp pagp dot1x lldp lacp udld loam esmc elmi ptppd mmrp
mvrp
  xconnect x.x.x.x 10 encapsulation mpls
  remote link failure notification
```

- On Cisco NCS 4206 and 4216 Series Aggregation Services Routers, MACsec doesn't support AAA accounting.
- MACsec is supported up to line rate on each interface. However, the forwarding capability may be limited by the maximum system forwarding capability.
- MACsec is supported on PE label on Cisco NCS 4206 and 4216 routers of NCS4200-1T16G-PS interface module.
- MACsec configuration on Ether Channel (Link bundling) isn't supported.
- Any interface that is configured with MACsec can't be part of Ether Channel.
- If the MKA session is torn down because of key unwrap failure, re-configure the pre-shared key-based MKA session using MACsec configuration commands on the respective interfaces to bring up the MKA session.
- MACsec-configured on physical interface with Ethernet Virtual Circuits (EVC) isn't supported. The EAPoL frames get dropped in such cases.
- On Cisco NCS 4206 and 4216 Series Aggregation Services Routers, the following table lists the Gigabit Ethernet interface and the maximum number of peers that are supported per interface.

Gigabit Ethernet Interface	Peers per Interface
1G	8
10G	32

- When `macsec dot1q-in-clear` is enabled, the native VLAN isn't supported.

Information About WAN MACsec and MKA Support Enhancements

MACsec and MKA Overview

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. Only host facing links (links between network access devices and endpoint devices such as a PC or IP phone) can be secured using MACsec.

The 802.1AE encryption with MACsec Key Agreement (MKA) is supported on downlink ports for encryption between the routers or switches and host devices.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet.

To provide MACsec services over the LAN or Metro Ethernet, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

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). Because the switch is the authenticator, it is also the key server, generating a random 128-bit secure association key (SAK), which it sends it 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 switch sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAP-over-LAN (EAPOL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participants after 3 hearbeats (each hearbeat is of 2 seconds), peers are deleted from the live peer list For example, 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.

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.

Starting with Cisco IOS XE Release 17.8.1, full HA, Power on Self Test (POST) and double tag support are available on NCS4200-1T16G-PS interface module. The POST tests the hardware to verify that all components of the device are operational and present. In the double tagging (qinq tag) method, the VLAN tag simply adds another tag to the tagged packets that enter the network. The purpose is to expand the VLAN space by tagging the tagged packets, thus producing a “double-tagged” frame. The expanded VLAN space allows the service provider to provide certain services, such as Internet access on specific VLANs for specific customers, and yet still allows the service provider to provide other types of services for their other customers on other VLANs. The Single Sign-On (SSO) and IM Online Insertion and Removal (OIR) triggers preserve MKA sessions.

Starting from release 17.12.1, Cisco IOS XE supports MACSec with PTP with G8275.1 profile on different 1G interfaces within the same NCS4200-1T16G-PS Interface Module. It does not support MACsec with PTP with G8275.1 on the same interface.

This limitation is applicable only when MACsec is configured on 1G port and PTP on 1G port with G8275.1 profile.

Table 2: MACsec and PTP Support on NCS4200-1T16G-PS

MACsec Support with PTP	Same Interface	Different Interface
1G—MACsec and PTP with G8275.1 profile	Not supported	Supported
1G—MACsec and PTP with G8275.2 and G8265.1 profiles	Not supported	Not supported

Benefits of WAN MACsec and MKA Support Enhancements

- Support for both Copper SFP 10M and 100M speed for MACsec.
- Support for point-to-point (P2P) deployment models.

- Support for 128 bit and 256 bit Advanced Encryption Standard–Galois Counter Mode (AES-GCM) encryption for data packets.
- Support for 128 bit and 256 bit Advanced Encryption Standard-Cipher-based Message Authentication Code (AEC-CMAC) encryption for control packets.
- Support for VLAN tag in the clear option to enable Carrier Ethernet Service Multiplexing.
- Support for coexisting of MACsec and Non-MACsec subinterfaces.
- Support for configurable Extensible Authentication Protocol over LAN (EAPOL) destination address.
- Support for configurable option to change the EAPOL Ethernet type.
- Support for configurable replay protection window size to accommodate packet reordering in the service provider network.
- Support for MACsec stateless switchover. During route processor (RP) switchover on dual RP setup, there is a teardown of existing MACsec session and the session is re-negotiated/reinitiated automatically (stateless switchover). During this process, some traffic drop might occur for a few seconds.

Best Practices for Implementing WAN MACsec and MKA Support Enhancements

- Ensure basic Layer 2 Ethernet connectivity is established and verified before attempting to enable MACsec. Basic ping between the customer edge devices must work.
- When you are configuring MACsec for the first time, ensure that you have out of band connectivity to the remote site to avoid locking yourself out after enabling MACsec, if the session fails to establish.
- We recommend that you configure an interface MTU, adjusting it for MACsec overhead, for example, 32 bytes. Although MACsec encryption and decryption occurs at the physical level and MTU size does not effect the source or destination router, it may effect the intermediate service provider router. Configuring an MTU value at the interface allows for MTU negotiation that includes MACsec overhead.

MKA Policy Inheritance

On WAN routers, MKA policy is inherited and also it has a default value. When a new session is started, the following rules apply:

- If an MKA policy is configured on a subinterface, it will be applied when an MKA session is started.
- If a MKA policy is not configured on a subinterface or physical interface, default policy is applied at session start.

Key Lifetime and Hitless Key Rollover

A MACsec key chain can have multiple pre-shared keys (PSK) each configured with a key ID and an optional lifetime. A key lifetime specifies when the key 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 key chain after the lifetime is expired. Time zone of the key can be local or UTC. Default time zone is UTC.

Use the **key chain** *name* **macsec** command to configure the MACsec key chain.

The key rolls over to the next key within the same key chain by configuring a second key in the key chain 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, key rolls over without traffic interruption.

Note The lifetime of the keys needs to be overlapped in order to achieve hitless key rollover.

Encryption Algorithms for Protocol Packets

Cryptographic Algorithm selection for MKA control protocol packets encryption is as follows:

- Cryptographic Algorithm to encrypt MKA control protocol packets is configured as part of the key chain. There can be only one cryptographic algorithm configured per key chain.
- A key server uses the configured MKA cryptographic algorithm from the key chain that is used.
- All nonkey servers must use the same cryptographic algorithm as the key server.

If an MKA cryptographic algorithm is not configured, a default cryptographic algorithm of AES-CMAC-128 (Cipher-based Message Authentication Code with 128-bit Advanced Encryption Standard) is used.

Encryption algorithm for Data packets:

```
mka policy p1
macsec-cipher-suite [gcm-aes-128 | gcm-aes-256
```

Encryption algorithm for MKA Control packets

```
key chain <name> macsec
key 01
key-string <Hex string>
cryptographic-algorithm [aes-256-cmac  | aes-128-cmac]
```

Replay Protection Window Size

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. Use the **macsec replay-protection window-size** command to change the replay window size. The range for window size is 0 to 4294967295.

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

WAN MACsec on Interface Module

The interface module NCS4200-1T16G-PS is supported on Cisco NCS 4206 and 4216 routers on 1GE and 10GE interfaces.

OIR Support

When the interface module is operationally inserted or removed (OIR), the configuration associated with that interface is preserved such that if the interface is ever reinserted into the system it appears with the same configuration. However, on Cisco NCS routers the following limitations apply for MACsec and MKA sessions:

- In some scale scenarios, after OIR MKA/MACsec session may be lost.
- MKA/MACsec session may be reestablished after OIR.

How to Configure WAN MACsec and MKA Support Enhancements

Configuring MKA

The MACsec Key Agreement (MKA) enables configuration and control of keying parameters. Perform the following task to configure MKA.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mka policy** *policy-name*

Example:

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

Configures an MKA policy.

Step 4 **include-icv-indicator**

Example:

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

(Optional) Include ICV indicator in MKPDU.

Step 5 **key-server priority** *key-server-priority*

Example:

```
Device(config-mka-policy)# key-server priority 200
```

(Optional) Configures MKA key server priority.

Step 6 **macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**}

Example:

```
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128 gcm-aes-256
```

(Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order.

Step 7 **sak-rekey interval** *interval*

Example:

```
Device(config-mka-policy)# sak-rekey interval 30
```

(Optional) Sets the SAK rekey interval (in seconds). The range is from 30 to 65535, and the default value is 0. The SAK rekey timer does not start by default until it is configured.

- To stop the SAK rekey timer, use the **no sak-rekey interval** command under the defined MKA policy.

Step 8 **confidentiality-offset 30**

Example:

```
Device(config-mka-policy)# confidentiality-offset 30
```

(Optional) Configures confidentiality offset for MACsec operation.

Step 9 **use-updated-eth-header**

(Optional) Enables interoperability with Cisco routers, and any port on a device that includes the updated ethernet header in MKPDUs for ICV calculation. This updated ethernet header is non-standard. Enabling this option ensures that an MKA session between the devices can be set up.

Before this fix, devices such as Cisco routers did not include the ethernet header for ICV calculation. With this command, an MKA session can be successfully established between your devices.

Step 10 **end**

Example:

```
Device(config-mka-policy)# end
```

Returns to privileged EXEC mode.

Example

You can use the **show mka policy** command to verify the configuration. Here's a sample output of the **show** command. If you do not want to include icv-indicator in MKPDUs, use the **no include-icv-indicator** command in the MKA policy.

```
MKA Policy Summary...

Codes : CO - Confidentiality Offset, ICVIND - Include ICV-Indicator,
        SAKR OLPL - SAK-Rekey On-Live-Peer-Loss,
        DP - Delay Protect, KS Prio - Key Server Priority

Policy            KS   DP    CO SAKR  ICVIND Cipher          Interfaces
Name              Prio          OLPL         Suite(s)        Applied
==============================================================================
*DEFAULT POLICY*  0    FALSE 0  FALSE TRUE   GCM-AES-128     N/A
                                             GCM-AES-256

confid50          0    FALSE 50 FALSE TRUE   GCM-AES-128
                                             GCM-AES-256

icv               0    FALSE 0  FALSE TRUE   GCM-AES-128     Te3/0/9
                                             GCM-AES-256

k10               0    FALSE 0  FALSE TRUE   GCM-AES-128
                                             GCM-AES-256
```

Configuring MKA Pre-shared Key

Perform the following task to configure MACsec Key Agreement (MKA) pre-shared key.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **key chain** *key-chain-name* [**macsec**]

Example:

```
Device(config)# Key chain keychain1 macsec
```

Configures a key chain and enters keychain configuration mode.

Step 4 **key** *hex-string*

Example:

```
Device(config-keychain)# key 9ABCD
```

Configures a key and enters keychain key configuration mode.

Step 5 **cryptographic-algorithm** {**aes-128-cmac** | **aes-256-cmac**}

Example:

```
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
```

Set cryptographic authentication algorithm.

Step 6 **key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}

Example:

```
Device(config-keychain-key)# key-string 0 pwd
```

Sets the password for a key string.

Step 7 **lifetime local** {{*day month year* **duration** *seconds*}

Example:

```
Device(config-keychain-key)# lifetime local 16:00:00 Nov 9 2014 duration 6000
```

Sets the lifetime for a key string. If you want infinite, then skip the configuration as the default value is infinite only.

The range you can specify for the duration is 1–864000 seconds.

Step 8 **end**

Example:

```
Device(config-keychain-key)# end
```

Returns to privileged EXEC mode.

Configuring MACsec and MKA on Interfaces

Perform the following task configure MACsec and MKA on an interface.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface GigabitEthernet 0/0/0
```

Enters interface configuration mode.

Step 4 **mka policy** *policy-name*

Example:

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

Configures an MKA policy

Step 5 **mka pre-shared-key key-chain** *key-chain-name*

Example:

```
Device(config-if)# mka pre-shared-key key-chain key-chain-name
```

Configures an MKA pre-shared-key key-chain keychain1

Note The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.

Step 6 **macsec**

Example:

```
Device(config-if)# macsec
```

Configures MACsec for the EAPOL frame ethernet type.

Step 7 **macsec replay-protection window-size**

Example:

```
Device(config-if)# macsec replay-protection window-size 10
```

Sets the MACsec window size for replay protection.

Step 8 **end**

Example:

```
Device(config-if)# end
```

Returns to privileged EXEC mode.

Configure WAN MACsec for QINQ Clear Tag

To configure WAN MACsec for QINQ clear tag:

```
configure terminal
!
```

```
key chain k10 macsec
key 10  cryptographic-algorithm aes-256-cmac
key-string  12345678901234567890123456789012123456789012345678901234567890123456789012!
mka policy p2
key-server priority 223
macsec-cipher-suite gcm-aes-256
!
interface TengigabitEthernet0/3/16
no ip address
ip mtu 1468
macsec dot1q-in-clear 2
service instance 200 ethernet
encapsulation dot1q 200 second-dot1q 201
rewrite ingress tag pop 2 symmetric
mka pre-shared-key key-chain k10
mka policy p2
macsec
bridge-domain 200
end
```

Verify POST Configuration

To verify the macsec Power on Self Test (POST) configuration, use the **show macsec post** command in privileged EXEC mode.

To verify the macsec Power on Self Test (POST) configuration:

```
 MACsec Capable Interface                          POST Result
-------------------------------------------------------------
 GigabitEthernet0/1/0                                 PASS
 GigabitEthernet0/1/2                                 PASS
 GigabitEthernet0/1/4                                 PASS
 GigabitEthernet0/1/6                                 PASS
 GigabitEthernet0/1/8                                 NONE
 GigabitEthernet0/1/10                               NONE
 GigabitEthernet0/1/12                               NONE
 GigabitEthernet0/1/14                               NONE
 TenGigabitEthernet0/1/16                          PASS
 GigabitEthernet0/2/0                                  PASS
GigabitEthernet0/2/2                                 PASS
 GigabitEthernet0/2/4                                  PASS
 GigabitEthernet0/2/6                                 NONE
```

MKA-PSK: CKN Behavior Change

For MKA-PSK sessions, instead of fixed 32 bytes, the Connectivity Association Key name (CKN) uses the same string as the CKN, which is configured as the hex-string for the key.

Example Configuration:

```
configure terminal
key chain abc macsec
  key 11
    cryptographic-algorithm aes-128-cmac
    key-string 1234567890123456789012345678901 3
    lifetime local 12:21:00 Sep 9 2015 infinite

 end
```

For the previous example, following is the **show** command output for the **show mka session** command:

```
Device# show mka session

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

=============================================================================================

           Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server

           Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN


=============================================================================================

           Et0/0          aabb.cc00.6600/0002     icv            NO                NO

           2              aabb.cc00.6500/0002 1                Secured           11   *Note
 that the CKN key-string is exactly the same that has been configured for the key as
hex-string.*
```

In the case of interoperability between two images -- one having the CKN behavior change, and one without the CKN behavior change, then the hex-string for the key must be a 64-character hex-string with zero padded to work on a device that has an image with the CKN behavior change. See the following example:

Configuration without CKN key-string behavior change:

```
      config t
        key chain abc macsec
         key 11
         cryptographic-algorithm aes-128-cmac
         key-string 12345678901234567890123456789013
         lifetime local 12:21:00 Sep 9 2015 1-864000
```

Configuration with CKN key-string behavior change:

```
      config t
        key chain abc macsec
         key 1100000000000000000000000000000000000000000000000000000000000000
         cryptographic-algorithm aes-128-cmac
         key-string 12345678901234567890123456789013
         lifetime local 12:21:00 Sep 9 2015 1-864000
```

Configuring an Option to Change the EAPoL Ethernet Type

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface GigabitEthernet 0/0/1
```

Enters interface configuration mode.

Step 4 **macsec** *eth-type*

Example:

```
Device(config)# macsec
```

Configures the Ethernet type for the Extensible Authentication Protocol (EAPoL) Frame.

Step 5 **eapol** *eth-type*

Example:

```
Device(config-if)# eapol eth-type 0xB860
```

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

Step 6 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for MACsec and MKA

Example: Point-to-point, CE to CE Connectivity Using EPL Service

The following is the sample configuration for point-to-point, Customer Edge to Customer Edge connectivity using Ethernet Private Line (EPL) using port-based service.

```
!Customer Edge 1
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.1 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*

!Customer Edge 2
key chain k1 macsec*
```

```
  key 01
  key-string 12345678901234567890123456789012
interface GigabitEthernet0/0/4
  ip address 10.3.1.2 255.255.255.0
  mka pre-shared-key key-chain k1*
  macsec*
```

Example: Point-to-point, CE to CE Connectivity Using EVPL Service

Example: Performing Maintenance Tasks Without Impacting Traffic

The following are sample configurations of performance maintenance tasks that do not impact traffic:

Changing a Pre-Shared Key (CAK Rollover)

The following is sample configuration for changing a pre-shared key:

Note Keys can be configured to automatically roll over to the next key by configuring a lifetime on both routers.

```
!From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012

!To
key chain k1 macsec*
  key 01
   key-string 12345678901234567890123456789012
   lifetime local 10:30:00 Oct 30 2014 11:30:00 Oct 30 2014
  key 02
   key-string 11145678901234567890123456789012
```

Changing a Key Chain (Keychain Rollover)

The following is the sample configuration for changing a key chain—Keychain Rollover

```
! From
key chain k1 macsec*
  key 01
  key-string 12345678901234567890123456789012
interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k1

! To
key chain k1 macsec
  key 01
   key-string 12345678901234567890123456789012
key chain k2 macsec
  key 02
   key-string abcdef0987654321abcdef0987654321
interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k2
```

Note The defined key ID, under any key chain, should be a unique value on the device.

A router can be become a key server by configuring a lower priority than other peer routers that participate in the same session. Configure a key server priority so that the key server selection is deterministic. For example, in a Hub and Spoke scenario, the most ideal place for a key server is the Hub site router.

```
!Hub Site (Key Server):
mka policy p1
key-server priority 0
!0 is the default.

interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k1
mka policy p1

!Spoke Sites (non-Key Servers):
mka policy p1
key-server priority 1

interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k1
mka policy p1
```

The following is sample configuration for changing Cipher Suite to encrypt data traffic:

```
mka policy p1
 macsec-cipher-suite gcm-aes-128
interface GigabitEthernet0/0/1
 mka policy p1

!Alternate configuration

mka policy p1
 macsec-cipher-suite gcm-aes-256
interface GigabitEthernet0/0/1
 mka policy p1
```

```
key chain k3 macsec
 key 01
   key-string abcdef0987654321abcdef0987654321
   cryptographic-algorithm aes-128-cmac
interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k3

!Alternate configuration:

key chain k3 macsec
 key 01
   key-string abcdef0987654321abcdef0987654321
   cryptographic-algorithm aes-256-cmac
interface TenGigabitEthernet0/0/0
 mka pre-shared-key key-chain k3
```

EAPOL Destination MAC address can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the subinterfaces, if configured at the physical interface level. To override the inherited value, configure the MAC address at the subinterface mode. Default EAPOL destination MAC address is 01:80:c2:00:00:03.

```
interface TenGigabitEthernet0/0/0
eapol destination-address <H.H.H>

!Alternate configuration

interface TenGigabitEthernet0/0/0
 bridge-group-address

!Alternate configuration

interface TenGigabitEthernet0/0/0
 lldp-multicast-address>

mka policy p1
 confidentiality-offset 30
interface GigabitEthernet0/0/1
 mka policy p1
```

Example: Performing Maintenance Tasks—Traffic Impacting

Changing a Replay Protection Window Size

Replay protection window can be changed from physical interface configuration mode or subinterface configuration mode and is automatically inherited by the subinterfaces if configured at the physical interface level. If you need to override the inherited value, configure it at the subinterface mode. However, MACsec on subinterfaces is not supported. The default replay protection window size is 64.

```
interface TenGigabitEthernet0/0/0
macsec replay-protection window-size 10

interface TenGigabitEthernet0/0/0
macsec replay-protection window-size 5
```

Enabling or Disabling VLAN (dot1q) Tag in the Clear Option

The **macsec dot1q-in-clear** command can only be configured on physical interface, and the setting is automatically inherited by the subinterfaces.

```
interface GigabitEthernet0/0/1
macsec dot1q-in-clear 1
```

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

```
interface GigabitEthernet0/0/1
macsec access-control must-secure/should-secure
```

Example: Configuring SyncE and MACSec

The process of enabling MACSec encryption for Synchronized Ethernet (SyncE) includes configurations to be performed in the global and interface configuration modes. MACSec is first configured in the global mode on the ingress and egress routers. Thereafter, MACSec and SyncE are configured on the required interfaces of the ingress and egress routers.

SyncE with Ethernet Synchronization Message Channel (ESMC) ensures network clock synchronization for successful network communication. The MACSec header is added to the ESMC packets. The ESMC MACSec header is used by SyncE interfaces to secure data on the physical media. See Configuring Synchronous Ethernet ESMC and SSM for more information.

This example shows how to configure MACSec in the global configuration mode:

At the ingress router:

```
Router(config)#key chain k10 macsec
Router(config-keychain-macsec)#key 10
Router(config-keychain-macsec-key)#cryptographic-algorithm aes-256-cmac
Router(config-keychain-macsec-key)#key-string
12345678901234567890123456789012123456789012345678901234567890123456789012
Router(config-keychain-macsec-key)#end
```

This example shows how to configure MACSec and SyncE on the GigabitEthernet0/5/0 interface:

```
Router(config)#interface GigabitEthernet0/5/0
Router(config-if)#mka pre-shared-key key-chain k10
Router(config-if)#macsec
MACSEC changed IP MTU on Gi0/5/0 from 1500 to 1468
Router(config-if)#no shutdown
Router(config-if)#synchronous mode
```

At the egress router:

This example shows how to configure MACSec in the global configuration mode:

```
Router(config)#key chain k10 macsec
Router(config-keychain-macsec)#key 10
Router(config-keychain-macsec-key)#cryptographic-algorithm aes-256-cmac
Router(config-keychain-macsec-key)#key-string
12345678901234567890123456789012123456789012345678901234567890123456789012
Router(config-keychain-macsec-key)#end
```

This example shows how to configure MACSec and SyncE on the GigabitEthernet0/2/0 interface:

```
Router(config)#interface GigabitEthernet0/2/0
Router(config-if)#mka pre-shared-key key-chain k10
Router(config-if)#macsec
MACSEC changed IP MTU on Gi0/5/0 from 1500 to 1468
Router(config-if)#no shutdown
Router(config-if)#synchronous mode
```

Verification

At the ingress router:

This example shows how to view the network clock synchronization details. The Gi0/2/0 interface denotes the egress router interface. Observe that SyncE and ESMC are enabled on the GigabitEthernet0/5/0 interface.

```
show network-clock synchronization detail
Symbols:     En - Enable, Dis - Disable, Adis - Admin Disable
             NA - Not Applicable
             *  - Synchronization source selected
             #  - Synchronization source force selected
             &  - Synchronization source manually switched
```

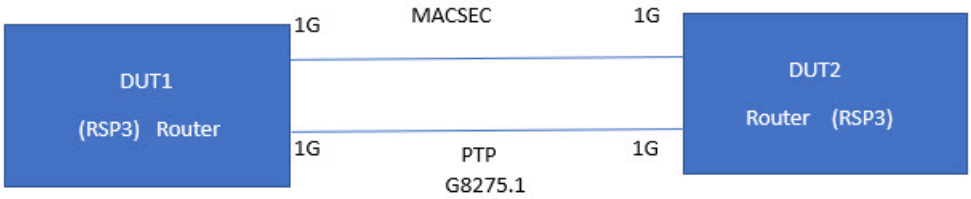
```
Automatic selection process : Enable
Equipment Clock : 2048 (EEC-Option1)
Clock State : Frequency Locked
Clock Mode : QL-Enable
ESMC : Enabled
SSM Option : 1
T0 : GigabitEthernet0/5/0
Hold-off (global) : 300 ms
Wait-to-restore (global) : 0 sec
Tsm Delay : 180 ms
Revertive : Yes
Force Switch: FALSE
Manual Switch: FALSE
Number of synchronization sources: 1
Squelch Threshold: QL-SEC
sm(netsync NETCLK_QL_ENABLE), running yes, state 1A
Last transition recorded: (begin)-> 2A (ql_mode_enable)-> 1A (src_added)-> 1A (sf_change)->
 1A (ql_change)-> 1A


Nominated Interfaces

 Interface            SigType     Mode/QL      Prio  QL_IN  ESMC Tx  ESMC Rx
 Internal             NA          NA/Dis       251   QL-SEC    NA        NA
*Gi0/2/0      NA             Sync/En         1      QL-PRC      -           -

Interface:
---------------------------------------------
Local Interface: Internal
Signal Type: NA
Mode: NA(Ql-enabled)
SSM Tx: DISABLED
SSM Rx: DISABLED
Priority: 251
QL Receive: QL-SEC
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: -
QL Transmit Configured: -
Hold-off: 0
Wait-to-restore: 0
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Active Alarms: None
Slot Disabled: FALSE
SNMP input source index: 1
SNMP parent list index: 0
Description: None

Local Interface: Gi0/5/0
Signal Type: NA
Mode: Synchronous(Ql-enabled)
ESMC Tx: ENABLED
ESMC Rx: ENABLED
Priority: 1
QL Receive: QL-PRC
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: QL-DNU
QL Transmit Configured: -
```

```
Hold-off: 300
Wait-to-restore: 0
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Active Alarms :  None
Slot Disabled: FALSE
SNMP input source index: 6
SNMP parent list index: 0
Description: None
Router#
--- 00:57:29 ---
show esmc detail | sec GigabitEthernet0/5/0
Interface: GigabitEthernet0/5/0
  Administrative configurations:
    Mode: Synchronous
    ESMC TX: Enable
    ESMC RX: Enable
    QL TX: -
    QL RX: -
  Operational status:
    Port status: UP
    QL Receive: QL-PRC
    QL Transmit: QL-DNU
    QL rx overrided: -
    ESMC Information rate: 1 packet/second
    ESMC Expiry: 5 second
    ESMC Tx Timer: Running
    ESMC Rx Timer: Running
    ESMC Tx interval count: 1
    ESMC INFO pkts in: 3034
    ESMC INFO pkts out: 3058
    ESMC EVENT pkts in: 4
    ESMC EVENT pkts out: 7
```

Example: Configuring MACsec and PTP

Scenario 1—Sample Topology for MACsec and PTP on Different 1G Interfaces

The MACsec is configured first at the global level and then at the interface level. PTP is configured first in synchronous mode at the interface level and then at the global level.



Configuring MACsec

```
Router#configure terminal
```

```
Router(config)#key chain k10 macsec
Router(config)#key 10
Router(config)#cryptographic-algorithm aes-256-cmac
Router(config)#key-string 12345678901234567890123456789012123456789012345678901234567890123456789012
Router(config)#interface GigabitEthernet 0/5/6
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#mka pre-shared-key key-chain k10
Router(config-if)#macsec
Router(config-if)#end
```

Verifying MACsec Configuration

```
Router#show mka sess inter Gi0/5/6

Summary of All Currently Active MKA Sessions on Interface GigabitEthernet0/5/6...
==========================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server

Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN

==========================================================================================
Gi0/5/6        00c1.b126.1c22/0028  *DEFAULT POLICY*  NO                 NO

40              0022.bddd.d97d/0033  1                 Secured            10

Router#
```

Configuring PTP (G8275.1)

```
Router#conf ter
Router(config)#interface GigabitEthernet0/5/4
Router(config-if)#synchronous mode
Router(config-if)#exit
Router(config)#network-clock revertive
Router(config)#network-clock synchronization automatic
Router(config)#network-clock synchronization mode QL-enabled
Router(config)#network-clock input source 1 External R0 10m
Router(config)#network-clock wait-to-restore 10 global
Router(config)#ptp clock ordinary domain 24  profile g8275.1
Router(config-ptp-clk)#tod R0 cisco
Router(config-ptp-clk)#input 1pps R0
Router(config-ptp-clk)#clock-port master master
Router(config-ptp-port)#transport ethernet multicast interface Gi0/3/4
Router(config-ptp-port)#end
```

Verifying PTP (G8275.1) Configuration

```
Router#show ptp clock  running

                      PTP Ordinary Clock [Domain 24]

         State          Ports          Pkts sent      Pkts rcvd      Redundancy Mode
         FREQ_LOCKED    1   1345           528              Hot standby

                               PORT SUMMARY

                     PTP Master
Name   Tx Mode      Role         Transport    State         Sessions     Port Addr
master mcast        master       Ethernet     Master        1            -
Router#
```