WAN MACSEC and MKA Support Enhancements

The WAN MACsec and MKA feature introduces MACsec support on WAN, and uplink support and Pre-shared key support for the Macsec Key Agreement protocol (MKA).

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 WAN, 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 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.

Benefits of WAN MACsec and MKA Support Enhancements

- Support for Point-to-point (P2P) deployment models.
- Support for Point-to-Multipoint (P2MP) deployment models.
- Support for multiple P2P and P2MP deployments on the same physical interface.
- Support for 128- and 256-bit Advanced Encryption Standard–Galois Counter Mode (AES-GCM) encryption for data packets.
- Support for 128- 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.

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 WAN 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 the **access-control should-secure** command while enabling MACsec for the first time and subsequently remove the command to change to default **access-control must-secure** , once the session establishment is successful, unless it is needed for migration.
- 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 is 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 an MKA policy is not configured on a subinterface, a policy that is configured on the physical interface is be applied at session start.
- 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 at which time 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** to configure the MACsec key chain.

You can 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 need 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]
```

It is recommended to change data packets cipher suite in the key server for the cipher suite rollover to be seamless, if the nonkey servers have the same cipher-suite configured in the list or is with default configuration.

Access Control Option for Smoother Migration

When MACsec is enabled on an interface, the entire interface traffic is secured, by default. MACsec does not allow any unencrypted packets to be transmitted or received from the same physical interface. However, to enable MACsec on selected subinterfaces, an additional Cisco proprietary extension has been implemented to allow unencrypted packets to be transmitted or received from the same physical interface.

Use the **macsec access-control** {**must-secure** | **should-secure**} command to control the behavior of unencrypted packets.

- The **should-secure** keyword allows unencrypted packets from the physical interface or subinterfaces to be transmitted or received.
- The **must-secure** keyword does not allow unencrypted packets from physical interface or subinterfaces to be transmitted or received. All such packets are dropped except for MKA control protocol packets
- If MACsec is enabled only on selected subinterfaces, configure the **should-secure** keyword option on the corresponding interface.

The default configuration for MACsec on subinterfaces is **macsec access-control must-secure** . This option is enabled by default when the **macsec** command is configured on an interface.

Note The **macsec access-control should-secure** command can be configured only at the interface level and not the subinterface. Configuring this command allows unencrypted traffic on a secured MACsec session.

Note For non-MACsec subinterface, you must configure the **should-secure** option for traffic to pass.

Extensible Authentication Protocol over LAN Destination Address

Before establishing a MACsec secure session, MKA (MACsec Key Agreement) is used as the control protocol. MKA selects the cipher suite to be used for encryption and to exchange the required keys and parameters between peers.

MKA uses Extensible Authentication Protocol over LAN (EAPoL) as the transport protocol to transmit MKA messages. By default, EAPoL uses a destination multicast MAC address of 01:80:c2:00:00:03 to multicast packets to multiple destinations. EAPoL is a standards-based protocol and other authentication mechanisms

such as IEEE 802.1X also use the same protocol. Devices in the service provider cloud might consume this packet (based on the destination multicast MAC address), and try to process the EAPoL packet and eventually drop the packet. This causes MKA session to fail.

Use the **eapol destination-address** command to change the destination MAC address of an EAPoL packet that is transmitted on an interface towards the service provider. This ensures that the service provider tunnels the packet like any other data packet instead of consuming them.

Note The EAPoL destination address can be configured independently on either physical or subinterface level. If it is configured on the physical interface, it is automatically inherited by the subinterfaces. Explicit configuration on the subinterface overrides the inherited value or policy for that subinterface.

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.

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.

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

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**mka policy** *policy-name* **Example:** `Router(config)# mka policy MKAPolicy`	Configures an MKA policy.
Step 4	**include-icv-indicator** **Example:** `Device(config-mka-policy)# include-icv-indicator`	(Optional) Includes ICV indicator in MKPDU.
Step 5	**key-server priority** *key-server-priority* **Example:** `Router(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:** `Router(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** *value* **Example:** `Router(config-mka-policy)# confidentiality-offset 30`	(Optional) Configures confidentiality offset for MACsec operation.
Step 9	**end** **Example:** `Router(config-mka-policy)# end`	Returns to privileged EXEC mode.

Configuring MKA Pre-Shared Key

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**Key chain** *key-chain-name* [**macsec**] **Example:** `Router(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode.
Step 4	**key** *hex string* **Example:** `Router(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode.
Step 5	**cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**} **Example:** `Router(config-keychain-key)# cryptographic-algorithm gcm-aes-128`	Set cryptographic authentication algorithm.
Step 6	**key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*} **Example:** `Router(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. The range you can specify for the duration is between 1 and 864000 seconds.
Step 8	**end** **Example:** `Router(config-keychain-key)# end`	Returns to privileged EXEC mode.

Configuring an Option to Change the EAPoL Ethernet Type

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.
Step 4	**eapol** *eth-type* **Example:** `Router(config-if)# eapol eth-type 876F`	Configures an ethernet type (Hexadecimal) for the EAPoL Frame on the interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Sample Configuration for Point-to-Point WAN MACsec

Example: Port Based WAN MACsec

Device1:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   ip address 1.1.1.1 255.255.255.0
   mka policy test
   mka pre-shared-key key-chain test128
   macsec
   exit
```

Device2:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   ip address 1.1.1.2 255.255.255.0
   mka policy test
   mka pre-shared-key key-chain test128
   macsec
   exit
```

Example: VLAN Based WAN MACsec

Device1:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   eapol destination-address broadcast-address
   mka policy test
   macsec dot1q-in-clear 1

   interface GigabitEthernet0/0/0.1
   encapsulation dot1q 10
   ip address 1.1.1.1 255.255.255.0
   mka pre-shared-key key-chain test128
    macsec
    exit
```

Device2:

```
configure terminal
    key chain test128 macsec
    key 1111
   cryptographic-algorithm aes-128-cmac
   key-string 11111111111111111111111111111111
   exit

   mka policy test
   macsec-cipher-suite gcm-aes-128 gcm-aes-256
   exit

   interface GigabitEthernet0/0/0
   eapol destination-address broadcast-address
   mka policy test
```

```
macsec dot1q-in-clear 1

interface GigabitEthernet0/0/0.1
encapsulation dot1q 10
ip address 1.1.1.2 255.255.255.0
mka pre-shared-key key-chain test128
macsec
exit
```

Sample Show Command Output for Port Based WAN MACsec

```
#show mka sessions

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

=============================================================================================
Gi0/0/0         b08b.d071.86a0/0007  test              NO                 YES

7               b08b.d079.8a10/0007  1                 Secured            1111



#show mka sessions detail

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

Local Tx-SCI.............. b08b.d071.86a0/0007
Interface MAC Address.... b08b.d071.86a0
MKA Port Identifier...... 7
Interface Name........... GigabitEthernet0/0/0
Audit Session ID.........
CAK Name (CKN)........... 1111
Member Identifier (MI)... 58D1ED5150ED8811970C4DB9
Message Number (MN)...... 18
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... 58D1ED5150ED8811970C4DB900000001 (1)
Old SAK Status........... FIRST-SAK
Old SAK AN............... 0
Old SAK KI (KN).......... FIRST-SAK (0)

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

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

```
Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

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

Live Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ------------------------------------------------------------------------------------------
  5225C3DE8DD5F35A3B55830C  53          b08b.d079.8a10/0007  0         YES            0

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

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

```
#show macsec statistics interface gigabitEthernet 0/0/0
MACsec Statistics for GigabitEthernet0/0/0
 SecY Counters
  Ingress Untag Pkts:       0
  Ingress No Tag Pkts:      0
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
  Ingress No SCI Pkts:      0
  Ingress Overrun Pkts:     0
  Ingress Validated Octets: 0
  Ingress Decrypted Octets: 966
  Egress Untag Pkts:        1
  Egress Too Long Pkts:     0
  Egress Protected Octets:  0
  Egress Encrypted Octets:  1387

 Controlled Port Counters
  IF In Octets:             1086
  IF In Packets:            10
  IF In Discard:            0
  IF In Errors:             0
  IF Out Octets:            1519
  IF Out Packets:           11
  IF Out Errors:            0

 Transmit SC Counters (SCI: B08BD07186A00007)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       11
 Transmit SA Counters (AN 0)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       11
```

```
 Receive SA Counters (SCI: B08BD0798A100007  AN 0)
  In Pkts Unchecked:        0
  In Pkts Delayed:          0
  In Pkts OK:               10
  In Pkts Invalid:          0
  In Pkts Not Valid:        0
  In Pkts Not using SA:     0
  In Pkts Unused SA:        0
  In Pkts Late:             0
```

Sample Show Command Output for VLAN Based WAN MACsec

```
#show mka sessions

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

=============================================================================================
Gi0/0/0.1       b08b.d071.86a0/0024  test              YES                YES

36              b08b.d079.8a10/001a  1                 Secured            1111



#show mka sessions detail

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

Local Tx-SCI.............. b08b.d071.86a0/0024
Interface MAC Address.... b08b.d071.86a0
MKA Port Identifier...... 36
Interface Name........... GigabitEthernet0/0/0.1
Audit Session ID.........
CAK Name (CKN)........... 1111
Member Identifier (MI)... 920AF42A7C0F5D2BDC0CBB99
Message Number (MN)...... 35
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 1
Latest SAK KI (KN)....... 920AF42A7C0F5D2BDC0CBB9900000002 (2)
Old SAK Status........... No Rx, No Tx
Old SAK AN............... 0
Old SAK KI (KN).......... RETIRED (1)

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

MKA Policy Name.......... test (inherited)
Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)
```

```
Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

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

Live Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ------------------------------------------------------------------------------------------
  E500D4C281BE01C8777901D4  18          b08b.d079.8a10/001a  0         YES            0

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

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

```
#show macsec statistics interface gigabitEthernet 0/0/0.1
MACsec Statistics for GigabitEthernet0/0/0.1
 SecY Counters
  Ingress Untag Pkts:       0
  Ingress No Tag Pkts:      0
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
  Ingress No SCI Pkts:      0
  Ingress Overrun Pkts:     0
  Ingress Validated Octets: 0
  Ingress Decrypted Octets: 4038
  Egress Untag Pkts:        1
  Egress Too Long Pkts:     0
  Egress Protected Octets:  0
  Egress Encrypted Octets:  4476

 Controlled Port Counters
  IF In Octets:             4926
  IF In Packets:            74
  IF In Discard:            0
  IF In Errors:             0
  IF Out Octets:            5460
  IF Out Packets:           82
  IF Out Errors:            0

 Transmit SC Counters (SCI: B08BD07186A00024)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       82
 Transmit SA Counters (AN 1)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       82

 Receive SA Counters (SCI: B08BD0798A10001A  AN 1)
```

```
In Pkts Unchecked:         0
In Pkts Delayed:           0
In Pkts OK:                74
In Pkts Invalid:           0
In Pkts Not Valid:         0
In Pkts Not using SA:      0
In Pkts Unused SA:         0
In Pkts Late:              0
```