MACsec

Media Access Control Security (MACsec) is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices.

For information about MACsec, including details about MACsec and MACsec Key Agreement (MKA), how to configure MKA and MACsec, and how to configure Cisco TrustSec MACsec, see Configuring MACsec Encryption.

This chapter includes the following information about MACsec specific to the IE 4000, IE 4010, and IE 5000 switches:

- PSK Based MKA Support for MACsec, page 243
- Certificate-based MACsec Encryption, page 247

Note: On the IE 4000, IE 4010, and the IE 5000, MACsec is included in the IP Services image only.

Guidelines and Limitations

MACsec on the IE5000 has the following guidelines and limitations:

- Both models of IE 5000 downlinks are fully interoperable with IE 4000, IE 4010, Catalyst 9300/3850, and Catalyst IE 3x00 platforms.
- On the IE-5000-16S12P, uplinks are fully functional when connected to another IE-5000-16S12P or a Catalyst 3850.
- On the IE-5000-12S12P-10G, uplinks when running at 10GE are fully functional when connected to another IE-5000-12S12P-10G running at 10GE or to a Catalyst 3850 running at 10GE.
- When an IE 5000 uplink is connected to a Catalyst 9300, the IE 5000 must be the key server. **CSCvs36043**
- IE-5000-12S12P-10G uplinks MACsec is not currently supported at GE speeds. **CSCvs41335**
- IE-5000-16S12P uplinks connected to downlinks of the IE 5000 and IE 4000 is not currently supported. **CSCvs44292**

MKA-PSK: CKN Behavior Change

To interoperate with Cisco switches running IOS XE, the CKN configuration must be zero-padded. From Cisco IOS XE Everest Release 16.6.1 onwards, for MKA-PSK sessions, instead of fixed 32 bytes, the Connectivity Association Key name (CKN) uses exactly the same string as the CKN, which is configured as the hex-string for the key.

Example configuration:

```
configure terminal
 key chain KEYCHAINONE macsec
 key 1234
   cryptographic-algorithm aes-128-cmac
   key-string 123456789ABCDEF0123456789ABCDEF0
   lifetime local 12:21:00 Sep 9 2015 infinite
 end
```

For the above example, following is the 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
|
==========================================================================
Gi1/1         34c0.f983.6c81/0001      POLICYONE       NO              YES
1             54a2.7498.5b01/0001      1               Secured         1234
```

Note that the CKN key-string is exactly the same that has been configured for the key as hex-string.

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

Configuration without CKN key-string behavior change:

```
config t
key chain KEYCHAINONE macsec
 key 1234
   cryptographic-algorithm aes-128-cmac
   key-string 123456789ABCDEF0123456789ABCDEF0
   lifetime local 12:21:00 Sep 9 2015 infinite
```

Output:

```
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
==========================================================================
Gi1/1         4c0.f983.6c81/0001       POLICYONE       NO              YES
1             54a2.7498.5b01/0001      1               Secured         123400000000000
                                                                       0000000000000000
                                                                       0000000000000000
                                                                       0000000000000000
```

Configuration with CKN key-string behavior change:

```
config t
key chain KEYCHAINONE macsec
 key 1234000000000000000000000000000000000000000000000000000000000000
   cryptographic-algorithm aes-128-cmac
   key-string 123456789ABCDEF0123456789ABCDEF0
   lifetime local 12:21:00 Sep 9 2015 infinite
```

Output:

```
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
=========================================================================
Gi1/1        34c0.f983.6c81/0001  POLICYONE       NO             YES
1            54a2.7498.5b01/0001  1               Secured        12340000000000000
                                                                 00000000000000000000
                                                                 000000000000000
                                                                 000000000000
```

PSK Based MKA Support for MACsec

This section provides information about configuring pre-shared key (PSK) based MACsec Key Agreement (MKA) MACsec encryption on the switch. This feature applies to Cisco IOS Release 15.2(7)E1a and later.

Information about PSK Based MKA

IE switches support Pairwise Master Key (PMK) Security Association Protocol (SAP) based support for MACsec to interconnect links between the switches. The PMK keys can be either derived statically from the switch configuration (manual mode) or derived from the RADIUS server during dot1X negotiation (dynamic mode). Manual mode does not support switch-to-host MACsec connections because SAP is a Cisco proprietary protocol.

IE switches have MKA support for MACSec on switch-to-host links. Here the keys are derived from the RADIUS server after dot1x authentication. However, manually configured PSK keys were not supported on IE switch platforms (running Cisco IOS) prior to Cisco IOS Release 15.2(7)E1a. Catalyst IE 3x00 platforms (running Cisco IOS XE) have PSK based MKA support for MACsec for statically derived keys from the switch configuration for switch-to-switch connections as well as dynamically derived keys from RADIUS server for switch-to-host links.

Catalyst IE 3x00 platforms do not have PMK SAP based support for MACsec. Therefore, for interoperability with the Catalyst IE 3x00 platforms, the PSK functionality is added to MACsec for Cisco IOS based IE switches.

Configuring PSK Based MKA

Follow the procedures in this section to configure PSK based MKA on IE 4000, IE 4010, and IE 5000 switches.

Configuring MKA

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

	Command	Purpose
1.	**enable** Example: `Device> enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	**configure terminal** Example: `Device# configure terminal`	Enters global configuration mode.
3.	**mka policy** *policy-name* Example: `Device(config)# mka policy MKAPolicy`	Configures an MKA policy.
4.	**key-server priority** *key-server-priority* Example: `Device(config-mka-policy)# key-server priority 200`	(Optional) Configures MKA key server priority.
5.	**macsec-cipher-suite {gcm-aes-128 }** Example: `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	(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.
6.	**replay-protection** Example: `Device(config-mka-policy)# replay-protection`	(Optional) Configure MKA to use replay protection for MACsec operation.
7.	**confidentiality-offset 30** Example: `Device(config-mka-policy)# confidentiality-offset 30`	(Optional) Configures confidentiality offset for MACsec operation.
8.	**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.

```
MKA Policy Summary...

Policy            KS          Delay     Replay    Window    Conf      Cipher       Interfaces
Name              Priority    Protect   Protect   Size      Offset    Suite(s)     Applied
================================================================================================
*DEFAULT POLICY*  0           FALSE     TRUE      0         0         CM-AES-128
POLICYONE         0           FALSE     TRUE      10        0         GCM-AES-128  Te1/26
```

Configuring MACsec and MKA on Interfaces

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

	Command	Purpose
1.	**enable** Example: `Device> enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	**configure terminal** Example: `Device# configure terminal`	Enters global configuration mode.
3.	**interface** *type number* Example: `Device(config)# interface GigabitEthernet 1/1`	Enters interface configuration mode.
4.	**mka policy** *policy-name* Example: `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy.
5.	**mka pre-shared-key key-chain** *key-chain-name* Example: `Device(config-if)# mka pre-shared-key key-chain keychain1`	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.

	Command	Purpose
6.	**macsec network-link** Example: `Device(config-if)#macsec network-link`	Configures PSK MKA MACsec on this interface. This is mutually exclusive with macsec.
7.	**macsec replay-protection window-size** Example: `Device(config-if)# macsec replay-protection window-size 10`	Sets the MACsec window size for replay protection.
8.	**end** Example: `Device(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.

	Command	Purpose
1.	**enable** Example: `Device> enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	**configure terminal** Example: `Device# configure terminal`	Enters global configuration mode.
3.	**key chain** *key-chain-name* [**macsec**] Example: `Device(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode
4.	**key** *hex-string* Example: `Device(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode.
5.	**cryptographic-algorithm {gcm-aes-128 }** Example: `Device(config-keychain-key)# cryptographic-algorithm gcm-aes-128`	Set cryptographic authentication algorithm.

	Command	Purpose
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.
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.
8.	**end** Example: `Device(config-mka-policy)# end`	Returns to privileged EXEC mode.

Certificate-based MACsec Encryption

This section provides information about Certificate-based MACsec Encryption. This feature applies to Cisco IOS Release 15.2(8)E and later.

Prerequisites for Certificate-based MACsec Encryption

- Certificate-based MACsec Encryption is supported on the IE4000, IE4010, and IE5000.
- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0. Refer to the Cisco Identity Services Engine Administrator Guide, Release 2.3.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.
- Ensure that 802.1x authentication and AAA are configured on your device.

Restrictions for Certificate-based MACsec Encryption

- MKA is not supported on port-channels.
- High Availability for MKA is not supported.
- When you remove **dot1x pae both** from an interface, all configuration related to dot1x is removed from the interface.
- Certificate-based MACsec is supported only if the access-session host-mode is configured in multiple-host mode. The other configuration modes (multi-auth, multi-domain, or single-host) are not supported.

Information About Certificate-based MACsec Encryption

MKA MACsec is supported on switch-to-switch links. Using IEEE 802.1X Port-based Authentication with Extensible Authentication Protocol (EAP-TLS), you can configure MKA MACsec between device ports. EAP-TLS allows mutual authentication and obtains an MSK (master session key) from which the connectivity association key (CAK) is derived for MKA protocol. Device certificates are carried, using EAP-TLS, for authentication to the AAA server.

Refer to Certificate-based MACsec Encryption For more information about Certificate-based MACsec Encryption, including how to configure Certificate-based MACsec Encryption using Remote Authentication.

Configuring Certificate-based MACsec Encryption using Remote Authentication

Follow these procedures to configure MACsec encryption using remote authentication:

- Configure Certificate Enrollment Manually
- Configure an Authentication Policy
- Configure EAP-TLS Profiles and IEEE 802.1x Credentials
- Configure MKA MACsec using EAP-TLS on Interfaces

Configuring Certificate Enrollment Manually

If network connection between the router and CA is not possible, perform the following task to set up manual certificate enrollment:

	Command or Action	Purpose
1.	`enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	`configure terminal`	Enters global configuration mode.
3.	**`crypto pki trustpoint`** *`server name`*	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
4.	`enrollment terminal`	Enroll via the terminal (cut-and-paste).
5.	**`rsakeypair`** *`label`*	Specifies which key pair to associate with the certificate.
6.	`serial-number`	Specifies the router serial number in the certificate request.
7.	**`Subject-name`** *`Line`*	Declares the subject name. For example: subject-name cn=MUSTS.mkadt.cisco.com ,OU=CSG Security,O=Cisco Systems,L=Bengaluru,ST=KA,C=IN

8.	**subject-alt-name** *Line*	include subject alternative name.
9.	**fqdn** *Line*	include fully-qualified domain name.
10.	`revocation-check none`	The **none** keyword specifies to ignore revocation check.
11.	`exit`	Exits global configuration mode.
12.	**crypto pki authenticate** *name*	Retrieves the CA certificate and authenticates it.
13.	**crypto pki enroll** *name*	Generates certificate request and displays the request for copying and pasting into the certificate server. Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
14.	**crypto pki import** *name* `certificate`	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch. **Note:** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
15.	`exit`	Exits global configuration mode.
16.	`show crypto pki certificate trustpoint name`	Displays information about the certificate for the trust point.
17.	`copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling 802.1x Authentication and Configuring AAA

	Command or Action	Purpose
1.	`enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	`configure terminal`	Enters global configuration mode.
3.	`aaa new-model`	Enables AAA.
4.	`dot1x system-auth-control`	Enables 802.1X on your device.
5.	**radius server** *name*	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
6.	**address** *ip-address* **auth-port** *port-number* **acct-port** *port-number*	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
7.	**automate-tester username** *username*	Enables the automated testing feature for the RADIUS server. With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks for a RADIUS response from the server. A success message is not necessary – a failed authentication suffices, because it shows that the server is alive.
8.	**key** *string*	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
9.	**radius-server deadtime** *minutes*	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
10.	`exit`	Returns to global configuration mode.
11.	**aaa group server radius** *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
12.	**server** *name*	Assigns the RADIUS server name.
13.	`exit`	Returns to global configuration mode.
14.	**aaa authentication dot1x default group** *group-name*	Sets the default authentication server group for IEEE 802.1x.
15.	**aaa authorization network default group** *group-name*	Sets the network authorization default group.

Configuring EAP-TLS Profile and 802.1x Credentials

	Command or Action	Purpose
1.	`enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	`configure terminal`	Enters global configuration mode.
3.	**`eap profile`** *`profile-name`*	Configures EAP profile and enters EAP profile configuration mode.
4.	`method tls`	Enables EAP-TLS method on the device.
5.	**`pki-trustpoint`** *`name`*	Sets the default PKI trustpoint.
6.	`exit`	Returns to global configuration mode.
7.	**`dot1x credentials`** *`profile-name`*	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
8.	**`username`** *`username`*	Sets the authentication user ID.
9.	`end`	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, perform the following task:

	Command or Action	Purpose
1.	`enable`	Enables privileged EXEC mode. ■ Enter your password if prompted.
2.	`configure terminal`	Enters global configuration mode.
3.	**`interface`** *`interface-id`*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
4.	`macsec network-link`	Enables MACsec on the interface.
5.	`authentication periodic`	Enables reauthentication for this port.
6.	`access-session host-mode multi-host`	Allows hosts to gain access to the interface.
7.	`access-session closed`	Prevents preauthentication access on the interface.
8.	`access-session port-control auto`	Sets the authorization state of a port.
9.	`dot1x pae both`	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
10.	`dot1x credentials profile`	Assigns a 802.1x credentials profile to the interface.

11.	**dot1x supplicant eap profile** *name*	Assigns the EAP-TLS profile to the interface.
	dot1x authenticator eap profile *name*	Assigns the EAP-TLS profile to the interface
12.	`service-policy type control subscriber control-policy name`	Applies a subscriber control policy to the interface.
13.	`exit`	Returns to privileged EXEC mode.
14.	`show macsec interface`	Displays MACsec details for the interface.
15.	`copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Verifying Certificate-based MACsec Encryption

Use the following **show** commands to verify the configuration of certificate-based MACsec encryption. Sample output is shown below.

```
Device#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
=============================================================================================
Gi1/18       7800.6750.0092/0012  *DEFAULT POLICY* NO             NO
18           5453.5632.0082/00021                  Secured        3E4CF3908A9055015FD95B890B94BFB5
```

The **show access-session interface** *interface-id* details displays detailed information about the access session for the given interface.

```
Device#show access-session interface gi 1/18 details
Interface:  GigabitEthernet1/18
          MAC Address:  5453.5632.0082
         IPv6 Address:  Unknown
         IPv4 Address:  Unknown
            User-Name:  scepen.mkadt.cisco.com
               Status:  Authorized
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
      Session timeout:  N/A
      Restart timeout:  N/A
Periodic Acct timeout:  N/A
       Session Uptime:  25s
    Common Session ID:  000000000000000C0011E814
      Acct Session ID:  0x00000001
               Handle:  0xC0000001
       Current Policy:  MUSTS_1
Local Policies:
        Service Template: DEFAULT_LINKSEC_POLICY_MUST_SECURE (priority 150)
      Security Policy:  Must Secure
      Security Status:  Link Secured
Server Policies:
Method status list:
      Method            State
      dot1xSupp          Authc Success
      dot1x              Authc Success
```

Configuration examples for Certificate-based MACsec Encryption

Example: Enrolling theCertificate

Configure Crypto PKI Trustpoint:

```
crypto pki trustpoint demo
 enrollment terminal
 serial-number
 fqdn MUSTS.mkadt.cisco.com
 subject-name cn=MUSTS.mkadt.cisco.com,OU=CSG Security,O=Cisco Systems,L=Bengaluru,ST=KA,C=IN
 subject-alt-name MUSTS.mkadt.cisco.com
 revocation-check none
 rsakeypair demo 2048
 !
```

Manual Installation of Root CA certificate:

```
crypto pki authenticate demo
```

Example: Enabling 802.1x Authentication and AAA Configuration

```
aaa new-model
dot1x system-auth-control
radius server ISE
address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646
key <secret configured on ise>
!
aaa group server radius ISEGRP
server name ISE
!
aaa authentication dot1x default group ISEGRP
aaa authorization network default group ISEGRP
!
```

Example: Configuring EAP-TLS Profile and 802.1X Credentials

```
eap profile scepen
 method tls
 pki-trustpoint demo
!
dot1x system-auth-control
dot1x credentials mis
 username scepen.mkadt.cisco.com
!
```

Example: Applying 802.1X, PKI, and MACsec Configuration on the Interface

```
interface GigabitEthernet1/2
 switchport mode access
 macsec network-link
 authentication periodic
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae both
 dot1x authenticator eap profile scepen
 dot1x credentials mis
 dot1x supplicant eap profile scepen
 service-policy type control subscriber MUSTS_1
```

```
!
```