Configuring MACsec Encryption

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

Security breaches can occur at any layer of the OSI model. Some of the common breaches at Layer 2 are MAC address spoofing, ARP spoofing, Denial of Service (DoS) attacks against a DHCP server, and VLAN hopping.

MACsec secures the data on physical media, making it impossible for data to be compromised at higher layers. As a result, MACsec encryption takes priority over any other encryption method for higher layers, such as IPsec and SSL.

MACsec provides encryption at the Layer 2, which is provided by the Advanced Encryption Standard (AES) algorithm that replaces the DES algorithm. MACsec uses the MACsec Key Agreement protocol (MKA) to exchange session keys, and manage encryption keys.

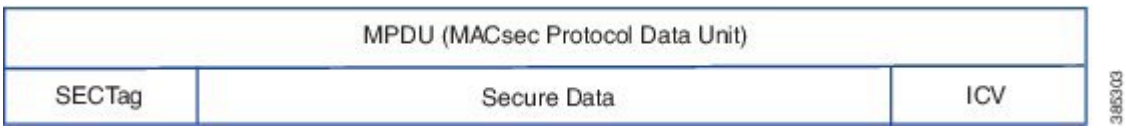
MACsec Frame Format

The MACsec header in a frame consists of three fields.

Table 1: Fields in MACsec Frame

Field	Size	Description
SECTag	8 or 16 bytes	Identifies the Security Association Key (SAK) to be used to validate the received frame. The security tag also provides replay protection when frames are received out of sequence. With Secure Channel Identifier (SCI) encoding, the security tag is 16 bytes in length, and without the encoding, 8 bytes in length (SCI encoding is optional).
Secure Data	2+ octets	Data in the frame that is encrypted using MACsec.
ICV	128 bit	Integrity Check Value (ICV) that provides the integrity check for the frame. Frames that do not match the expected ICV are dropped at the port.

Figure 1: MACsec Frame Format



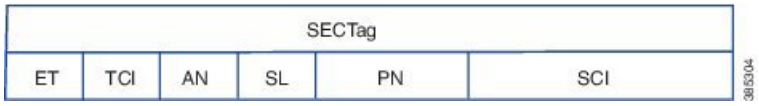
MACsec SECTag Format

The MACsec SECTag header in a frame consists of the following fields.

Table 2: Fields in MACsec SECTag Frame

Field	Size	Description
ET	16 bit	MACsec EtherType value (0x88E5) for MACsec packet.
TCI	6 bit	Tag control information that indicates how frame is protected.
AN	2 bit	Association number.
SL	8 bit	Short length of MAC service data unit (MSDU).
PN	32 bit	Packet sequence number.
SCI	64 bit	(optional) Secure channel identifier.

Figure 2: MACsec SECTag Frame Format



MACsec Key Agreement

The MACsec Key Agreement (MKA) Protocol, defined in IEEE 802.1X-2010, provides the required session keys and manages the required encryption keys. MKA is a multipoint to multipoint protocol that defines the mechanism to generate and distribute keys for MACsec.

MKA allows authorized multiple devices that possess secret key (CAK) to participate in a CA (Connectivity Association). It defines the election of Key Server (KS) that generates the Security Association Key(SAK) and distributes the SAK to all the participants. MACsec frames across the devices are secured using SAK. MKA also transports MACsec capability such as delay protection and confidentiality offset.

MKA operates in two modes.

- MKA using pre-shared key (PSK) authentication. See Configure MACsec Encryption Using PSK Authenication, on page 6 for configuration steps.
- MKA using Extensible Authentication Protocol (EAP) authentication. EAP mode uses 802.1X authentication. See Configure MACsec Encryption Using EAP-TLS Authentication, on page 36 for configuration steps.

MACsec in NCS 1002

MACsec in NCS 1002 has the following characteristics or limitations.

- Supports 256-bit Extended Packet Numbering (XPN) according to IEEE 802.1AEbn-2011.
- Supports GCM-AES-XPN-256 as the default cipher.
- Supports AES-128-CMAC and AES-256-CMAC cryptographic algorithms.
- Supports SecY function in the data plane as specified by IEEE 802.1 AE-2006 specification.
- Supports only 2 x 100G client and 1 x 200G trunk traffic.
- Supports only cumulative statistics for MACsec counters.
- Supported only with the ncs1k-k9sec package.
- Not supported in the headless mode.
- Recommended to upgrade the nodes to R6.2.1 and bring up the 100G MACsec sessions.
- For 100G MACsec deployed in R6.1.1 and R6.1.2: If the customer migrates from R6.1.2 to R6.2.1, traffic hit occurs. The subsequent headless operations will not have any traffic drops.

Note When the user needs to use the MACsec feature and upgrades from R6.0.1 to 6.1.1, the control FPGA (CTRL_BKP_UP, CTRL_BKP_LOW, CTRL_FPGA_UP, and CTRL_FPGA_LOW) needs to be upgraded to the latest firmware version provided by R6.1.1. See Verify Firmware Version for more information.

Supported Configurations in Encrypted Mode

The following configurations are supported on client and trunk ports in each slice configured in encrypted mode.

Client Ports	Trunk Ports
2 x 100G	1 x 200G
10 x 10G	1 x 100G
2 x 40G	1 x 100G

All the configurations can be accomplished using appropriate values for client bitrate and trunk bitrate parameters of the **hw-module** command.

The following table describes the client and trunk ports in slice 0 that are enabled or disabled for each supported configuration in encrypted mode.

Client Data Rate	Trunk Data Rate	Client Port 0	Client Port 1	Client Port 2	Client Port 3	Client Port 4	Trunk Port 5	Trunk Port 6
100G	200G	D	D	D	E	E	D	E

Client Data Rate	Trunk Data Rate	Client Port 0	Client Port 1	Client Port 2	Client Port 3	Client Port 4	Trunk Port 5	Trunk Port 6
10G	100G	D	D	Only the first and second controllers are active.	E	E	D	E
40G	100G	D	D	D	E	E	D	E

E indicates that the port is enabled; D indicates that the port is disabled.

Illustrations for Supported Configurations in Encrypted Mode

The following illustrations describe the mapping of traffic from client to trunk ports in encrypted mode for the supported configurations.

Figure 3: Client: 2 x 100G Trunk: 1 x 200G

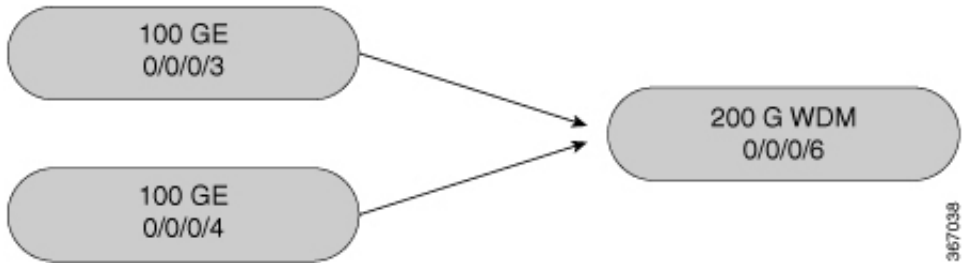


Figure 4: Client: 10 x 10G Trunk: 1 x 100G

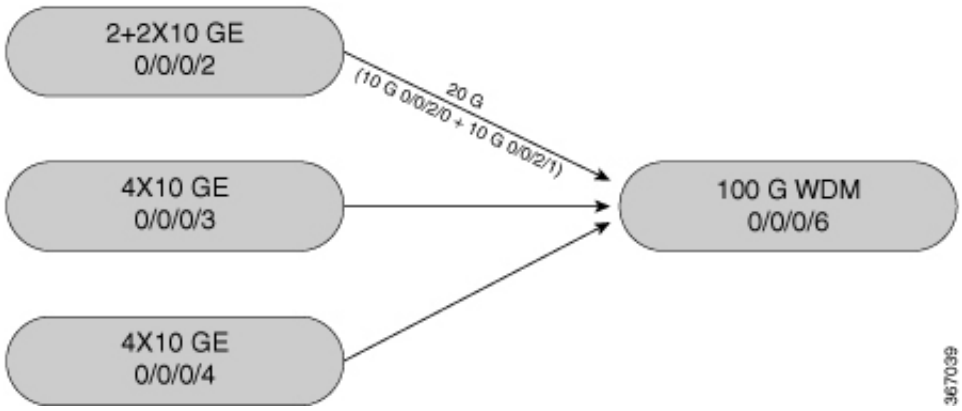
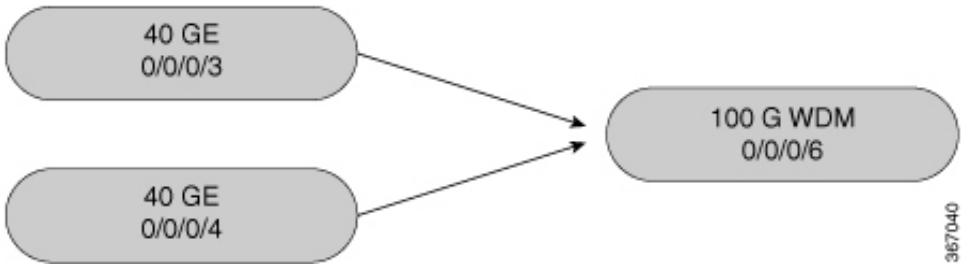


Figure 5: Client: 2 x 40G Trunk: 1 x 100G



Configure MACsec Encryption Using PSK Authenication

Configuring MACsec encryption using PSK authentication involves the following tasks:

1. Configure MACsec Key Chain, on page 7
2. Verify MACsec Key Chain, on page 8
3. Configure MACsec Policy, on page 9
4. Verify MACsec Policy, on page 11
5. Configure the Slice
6. Verify Slice Configuration
7. Apply MACsec Configuration on MACsec Controller, on page 19
8. Verify MACsec Configuration on MACsec Controller, on page 19

MACsec Key Chain

A MACsec key chain is a collection of keys used to authenticate peers needing to exchange encrypted information. While creating a key chain, we define the key(s), key string with password, the cryptographic algorithm, and the key lifetime.

- The key can be up to 64 characters in length.
- The key name must be of even number of characters. Entering an odd number of characters will exit the MACsec configuration mode. The key name must match on both the sides.
- The key string is 64 hexadecimal characters in length when AES 256-bit encryption algorithm is used and 32 hexadecimal characters in length when AES 128-bit encryption algorithm is used. It is recommended to create key name and provide the key-string and lifetime.
- The lifetime period (validity period of the key) can be configured, with a duration in seconds, as a validity period between two dates (for example, Jan 01 2016 to Dec 31 2016), or with infinite validity. The key is valid from the time you configure (in HH:MM:SS format). The duration is configured in seconds. The overlapping time must be configured in two keys to avoid traffic loss.
- The keys roll over to the next key within the same key chain by configuring a second key (key 02) in the key chain and configuring lifetime for the first key. When the lifetime of the first key (key 01) expires,

it automatically rolls over to the next key in the list. If the same key is configured simultaneously on both sides of the link, the key rollover is hitless and the key rolls over without interruption in traffic. Based on IEEE 802.1x, the overlapping time between the keys in a key chain can be up to 20 seconds. The re-key operation can take up to 16 seconds.

Configure MACsec Key Chain

configure

key chain *key-chain-name* **macsec**

key *key-name*

key-string *password* **cryptographic-algorithm** {**aes-256-cmac** | **aes-128-cmac**}

lifetime *start_time start_date* { *end_time end_date* | **duration** *validity* | **infinite** }

exit

commit

Examples

The following is a sample in which the key chain is configured with AES 256-bit encryption algorithm and specific duration for the lifetime period.

```
configure
key chain mac_chain macsec
key 1234abcd5678
key-string 12345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm aes-256-cmac
lifetime 05:00:00 01 july 2016 duration 1800
exit
commit
```

The following is a sample in which the key chain is configured with AES 256-bit encryption algorithm and defined period for the lifetime period.

```
configure
key chain mac_chain macsec
key 1234abcd5678
key-string 12345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm aes-256-cmac
lifetime 05:00:00 20 july 2016 12:00:00 30 september 2016
exit
commit
```

The following is a sample in which the key chain is configured with AES 256-bit encryption algorithm and infinite duration for the lifetime period.

```
configure
key chain mac_chain macsec
key 1234abcd5678
key-string 12345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm aes-256-cmac
lifetime 05:00:00 01 January 2015 infinite
```

```
exit
commit
```

The following is a sample in which the key chain is configured with AES 128-bit encryption algorithm and specific duration for the lifetime period.

```
configure
key chain mac_chain macsec
key abc1
key-string 12345678123456781234567812345678 cryptographic-algorithm aes-128-cmac
lifetime 17:30:00 31 August 2016 duration 4000
exit
commit
```

The following is a sample in which the key chain is configured with AES 128-bit encryption algorithm and defined period for the lifetime period.

```
configure
key chain mac_chain macsec
key abc2
key-string 12345678123456781234567812345678 cryptographic-algorithm aes-128-cmac
lifetime 17:30:00 31 August 2016 12:00:00 30 september 2016
exit
commit
```

The following is a sample in which the key chain is configured with AES 128-bit encryption algorithm and infinite duration for the lifetime period.

```
configure
key chain mac_chain macsec
key abc3
key-string 12345678123456781234567812345678 cryptographic-algorithm aes-128-cmac
lifetime 05:00:00 01 January 2015 infinite
exit
commit
```

Associated Commands

- key chain
- key
- key-string
- cryptographic-algorithm
- lifetime

Verify MACsec Key Chain

show key chain

```
Wed Aug 17 14:34:00.056 IST

Key-chain: TESTMA -(MacSec)

  Key BDA123
```

```
    Key-String -- 08701E1D5D4C53404A5A5E577E7E727F6B647040534355560E080A00005B554F4E
    Cryptographic-Algorithm -- ALG_AES_256_CMAC
    Send lifetime --  19:05:00, 16 Aug 2016 - Always valid  [Valid now]

Key-chain: mac_chain -(MacSec)

  Key abc1
    Key-String -- 12485744465E5A53727A767B676074445F475152020C0E040C5F514B420C0E000B
    Cryptographic-Algorithm -- ALG_AES_128_CMAC
    Send lifetime --  17:30:00, 31 Aug 2016 - (Duration) 4000

  Key abc2
    Key-String -- 135445415F59527D73757A60617745504E5253050D0D050356524A450D0D01040A
    Cryptographic-Algorithm -- ALG_AES_128_CMAC
    Send lifetime --  17:30:00, 31 Aug 2016 - 12:00:00, 30 Sep 2016

  Key abc3
    Key-String -- 101F5B4A5142445C54557878707D65627A4255455754000E0002065D574D400E00
    Cryptographic-Algorithm -- ALG_AES_128_CMAC
    Send lifetime --  05:00:00, 01 Jan 2015 - Always valid  [Valid now]
```

MACsec Policy

You apply a defined MACsec policy to enable MKA on the controller. You can configure these parameters for MACsec policy:

- Policy name, not to exceed 16 ASCII characters.
- Confidentiality (encryption) offset of 0 bytes.
- Replay protection. You can configure MACsec window size, as defined by the number of out-of-order frames that are accepted. This value is used while installing the security associations in the MACsec. A value of 0 means that frames are accepted only in the correct order.
- The cipher suite to be used for MACsec encryption is GCM-AES-XPN-256.
- The range of **key server priority** parameter is 0 to 255. Lower the value, higher the preference to be selected as the key server.
- The **security-policy** parameter configures the type of traffic (encrypted traffic or all traffic) that is allowed through the controller configured with MACsec. The default value of **security-policy** parameter is **must-secure** that indicates unencrypted packets cannot be transmitted or received except MKA control protocol packets.

Configure MACsec Policy

configure

macsec-policy *policy-name*

cipher-suite *encryption-suite*

conf-offset *offset-value*

key-server-priority *value*

security-policy {should-secure | must-secure}

window-size *value*

exit

commit

Examples

Example 1: The following is a sample of configuring the MACsec policy.

```
configure
macsec-policy mac_policy
cipher-suite GCM-AES-XPN-256
conf-offset CONF-OFFSET-0
key-server-priority 0
security-policy must-secure
window-size 64
exit
commit
```

Example 2: If a specific setting does not apply to NCS 1002, the setting is rejected during commit.

```
configure
macsec-policy mac_policy
vlan-tags-in-clear 1
commit


Thu Aug 4 19:31:38.033 UTC

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue
 'show configuration failed [inheritance]' from this session to view the errors

show configuration failed


Thu Aug 4 19:31:56.601 UTC
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.


macsec-policy mac_policy
!!% A verifier or EDM callback function returned: 'not supported': vlan_tags_in_clear is
not supported.

vlan-tags-in-clear 1
!!% A verifier or EDM callback function returned: 'not supported': vlan_tags_in_clear is
not supported.

!
end
```

Example 3: If a specific configuration in the batch operation is not supported, the entire batch is rejected during commit.

```
configure
macsec-policy mac_policy
cipher-suite GCM-AES-XPN-256
window-size 64
```

```
conf-offset CONF-OFFSET-0
commit


Thu Aug 4 19:37:22.355 UTC

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed [inheritance]'
from this session to view the errors

show configuration failed


Thu Aug 4 19:38:29.948 UTC
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.


macsec-policy mac_policy
!!% A verifier or EDM callback function returned: 'not supported': The only supported
conf_offset is CONF-OFFSET-0

conf-offset CONF-OFFSET-0
!!% A verifier or EDM callback function returned: 'not supported': The only supported
conf_offset is CONF-OFFSET-0

window-size 64
!!% A verifier or EDM callback function returned: 'not supported': The only supported
conf_offset is CONF-OFFSET-0

cipher-suite GCM-AES-XPN-256
!!% A verifier or EDM callback function returned: 'not supported': The only supported
conf_offset is CONF-OFFSET-0

!
end
```

Associated Commands

- macsec-policy
- cipher-suite
- conf-offset
- key-server-priority
- security-policy
- window-size

Verify MACsec Policy

show macsec policy

```
Sun Dec 18 14:22:23.587 IST

Total Number of Policies = 3
```

```
=================================================================
Policy            Cipher              Key-Svr   Window    Conf
name              Suite               Priority  Size      Offset
=================================================================
*DEFAULT POLICY*  GCM-AES-XPN-256     16        64        0
kcp1              GCM-AES-XPN-256     16        128       0
kcp2              GCM-AES-XPN-256     16        256       0
```

show macsec policy 5

```
Wed Mar 30 12:49:29.371 UTC
=================================================================
Policy            Cipher              Key-Svr   Window    Conf
name              Suite               Priority  Size      Offset
=================================================================
5                 GCM-AES-XPN-256     37        64        0
```

If the values you see are different from the ones you configured, then check your configuration by running the **show run macsec-policy** command.

MACsec Controllers

MACsec controllers are created when a slice is provisioned with the **encrypted** keyword. The MACsec controller is used to configure the MACsec parameters. All the MACsec statistics is available on the MACsec controller. The MACsec controller is represented in the *Rack/Slot/Instance/Port* format, for example, 0/0/0/3.

A unique MAC address is generated for each MACsec controller. When software is upgraded to R6.2.2 with traffic, traffic loss occurs for the slice configured in encrypted mode.

Configure the Slice

You can configure the slice with traffic on client and trunk ports. All five client ports of the slice need to be configured at the same bitrate except for mixed mode configuration. Both the trunk ports are always set with the same FEC mode. The slice can be configured to send encrypted traffic from R6.1.1.

See the Supported Configurations in Encrypted Mode, on page 4 section to determine the supported configurations on the client and trunk ports in each slice configured in encrypted mode

Note When the slice is configured in encrypted mode, the drop-lldp cannot be enabled.

Note When NCS 1002 is installed in a system where both the trunk interfaces in a slice are used, the two 250Gb 16QAM signals need to be co-routed on the same fiber (mandatory when the 5x100Gb client port is provisioned). Also, it is recommended to use adjacent wavelengths when the line modulation is set to 250Gb 16QAM. The reason for this is that the chromatic dispersion generates skew between wavelengths. Assuming a Dispersion of 10000 ps/nm, a span of 500 km, and using adjacent channel, the skew is evaluated in less than 200 ns and it is compensated by the deskew capability of NCS 1002. If the delta between the used channels is increased, the skew increases and it might exceed the skew compensation done by NCS 1002.

To configure the slice with unencrypted traffic, use the following commands.

configure

hw-module location *location* **slice** [*slice_number* | **all**] **client bitrate** { **10G** | **40G** | **100G** } **trunk bitrate** { **100G** | **200G** | **250G** } **fec** { **softdecision7** | **softdecision20** }

commit

To configure the slice with mixed mode, use the following commands.

configure

hw-module location *location* **slice** [*slice_number* | **all**] **client bitrate 10G-100G trunk bitrate 200G fec** { **softdecision7** | **softdecision20** }

commit

To configure the slice with encrypted traffic, use the following commands.

configure

hw-module location *location* **slice** [*slice_number* | **all**] **client bitrate** { **10G** | **40G** | **100G** } **trunk bitrate** { **100G** | **200G** } **fec** { **softdecision7** | **softdecision20** } [**encrypted**]

commit

Examples

The following is a sample in which slice 0 is configured in mixed mode, and FEC on the trunk ports is set to softdecision7.

```
configure
hw-module location 0/RP0/CPU0 slice 0 client bitrate 10G-100G trunk bitrate 200G fec
SoftDecision7
commit
```

The following is a sample in which slice 0 is configured to send encrypted traffic with 100G client rate, 200G trunk rate, and FEC on the trunk ports is set to softdecision7.

```
configure
hw-module location 0/RP0/CPU0 slice 0 client bitrate 100G trunk bitrate 200G softdecision7
 encrypted
commit
```

The following is a sample in which slice 0 is configured to send encrypted traffic with 10G client rate, 100G trunk rate, and FEC on the trunk ports is set to softdecision20. When a slice is configured with 10G client rate in encrypted mode, ten MACsec controllers are created for each slice. When all the four slices are configured with 10G client rate in encrypted mode, forty MACsec controllers are created for NCS 1002. Two MACsec controllers are created for the middle port, four controllers for the fourth port, and four controllers for the fifth port per slice.

```
configure
hw-module location 0/RP0/CPU0 slice 0 client bitrate 10G trunk bitrate 100G softdecision20
 encrypted
commit
```

The following is a sample in which slice 0 is configured to send encrypted traffic with 40G client rate, 100G trunk rate, and FEC on the trunk ports is set to softdecision20.

```
configure
hw-module location 0/RP0/CPU0 slice 0 client bitrate 40G trunk bitrate 100G softdecision20
 encrypted
commit
```

The following is a sample to configure all the slices with a specific client rate and trunk rate.

```
configure
hw-module location 0/RP0/CPU0 slice all client bitrate 10G trunk bitrate 100G fec
softDecision7
commit


configure
hw-module location 0/RP0/CPU0 slice all client bitrate 40G trunk bitrate 100G fec
softDecision7
commit


configure
hw-module location 0/RP0/CPU0 slice all client bitrate 100G trunk bitrate 200G fec
softDecision7
commit
```

The following is a sample to remove the configuration from all the slices.

```
configure
no hw-module location 0/RP0/CPU0 slice all client bitrate 10G trunk bitrate 100G fec
softDecision7
commit


configure
no hw-module location 0/RP0/CPU0 slice all client bitrate 40G trunk bitrate 100G fec
softDecision7
commit


configure
no hw-module location 0/RP0/CPU0 slice all client bitrate 100G trunk bitrate 200G fec
softDecision7
commit
```

Note Until R6.3.2, if the user wants to modify the slice configuration using the **hw-module** command, the existing slice must be deleted and new slice must be configured. From R6.5.1, the user can directly change the existing parameters on the configured slice without deleting the slice.

The slice configuration can be done using hw-module configuration or terminal-device configuration. However, the hw-module configuration cannot be modified or deleted using the terminal-device configuration and vice versa.

Example of Slice Modification

Display the slice configuration.

```
RP/0/RP0/CPU0:ios# show hw-module slice 0


Fri Jun  1 10:07:22.035 IST
Slice ID:               0
```

```
Status:                  Provisioned
Client Bitrate:          100
Trunk Bitrate:           200
DP FPGA FW Type:         X100
DP FPGA FW Version:      01.01
HW Status:               CURRENT

Encryption Supported:    FALSE
LLDP Drop Enabled:       FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/5   CoherentDSP0/0/0/6
Traffic Split Percentage

HundredGigECtrlr0/0/0/0                          100                   0
HundredGigECtrlr0/0/0/1                          100                   0
HundredGigECtrlr0/0/0/3                            0                 100
HundredGigECtrlr0/0/0/4                            0                 100
```

Modify the slice configuration with a different trunk bit rate without deleting the slice.

```
configure
hw-module location 0/RP0/CPU0 slice 0 client bitrate 100G trunk bitrate 250G fec
SoftDecision20
commit
end
```

Verify the slice re-configuration.

```
RP/0/RP0/CPU0:ios# show hw-module slice 0


Fri Jun  1 10:07:45.959 IST
Slice ID:                0
Status:                  Provisioning In Progress
Client Bitrate:          100
Trunk Bitrate:           250
DP FPGA FW Type:         UNKNOWN
DP FPGA FW Version:      00.00
HW Status:               CURRENT

Encryption Supported:    FALSE
LLDP Drop Enabled:       FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/5   CoherentDSP0/0/0/6
Traffic Split Percentage

HundredGigECtrlr0/0/0/0                          100                   0
HundredGigECtrlr0/0/0/1                          100                   0
HundredGigECtrlr0/0/0/2                           50                  50
HundredGigECtrlr0/0/0/3                            0                 100
HundredGigECtrlr0/0/0/4                            0                 100
```

Associated Commands

- hw-module
- show hw-module

Verify Slice Configuration

Use this procedure to verify whether the slice is correctly configured.

Procedure

show hw-module { **slice** [*slicenumber* | **all**] }

Example:

```
RP/0/RP0/CPU0:ios# show hw-module slice 0


Thu Aug 11 16:16:58.935 IST
Slice ID:                0
Status:                  Provisioned
Client Bitrate:          100
Trunk Bitrate:           200
DP FPGA FW Type:         M100
DP FPGA FW Version:      02.00
HW Status:               CURRENT

Encryption Supported:    TRUE
LLDP Drop Enabled:       FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/6
Traffic Split Percentage

HundredGigECtrlr0/0/0/3                  100
HundredGigECtrlr0/0/0/4                  100
```

```
RP/0/RP0/CPU0:ios# show hw-module slice 0


Sun Dec 18 13:59:18.805 IST
Slice ID:                0
Status:                  Provisioned
Client Bitrate:          40
Trunk Bitrate:           100
DP FPGA FW Type:         MM40
DP FPGA FW Version:      03.00
HW Status:               CURRENT

Encryption Supported:    TRUE
LLDP Drop Enabled:       FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/6
Traffic Split Percentage

FortyGigECtrlr0/0/0/3                    100
FortyGigECtrlr0/0/0/4                    100


RP/0/RP0/CPU0:ios# show hw-module slice 1


Tue Jan  1 06:55:12.293 UTC
Slice ID:                1
Status:                  Provisioned
Client Bitrate:          10
Trunk Bitrate:           100
DP FPGA FW Type:         MM10
DP FPGA FW Version:      03.00
HW Status:               CURRENT

Encryption Supported:    TRUE
LLDP Drop Enabled:       FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/13
Traffic Split Percentage
```

```
TenGigECtrlr0/0/0/9/1                           100
TenGigECtrlr0/0/0/9/2                           100
TenGigECtrlr0/0/0/10/1                          100
TenGigECtrlr0/0/0/10/2                          100
TenGigECtrlr0/0/0/10/3                          100
TenGigECtrlr0/0/0/10/4                          100
TenGigECtrlr0/0/0/11/1                          100
TenGigECtrlr0/0/0/11/2                          100
TenGigECtrlr0/0/0/11/3                          100
TenGigECtrlr0/0/0/11/4                          100

RP/0/RP0/CPU0:ios# show hw-module slice 2


Slice ID:               2
Status:                 Provisioned
Client Bitrate:         10,100
Trunk Bitrate:          200
DP FPGA FW Type:        RMM
DP FPGA FW Version:     04.00
HW Status:              CURRENT

Encryption Supported:   FALSE
LLDP Drop Enabled:      FALSE
Client Port -   Trunk Port          CoherentDSP0/0/0/19  CoherentDSP0/0/0/20
Traffic Split Percentage

HundredGigECtrlr0/0/0/14                        100                   0
HundredGigECtrlr0/0/0/15                        100                   0
HundredGigECtrlr0/0/0/16                          0                 100
TenGigECtrlr0/0/0/17/1                            0                 100
TenGigECtrlr0/0/0/17/2                            0                 100
TenGigECtrlr0/0/0/17/3                            0                 100
TenGigECtrlr0/0/0/17/4                            0                 100
TenGigECtrlr0/0/0/18/1                            0                 100
TenGigECtrlr0/0/0/18/2                            0                 100
TenGigECtrlr0/0/0/18/3                            0                 100
TenGigECtrlr0/0/0/18/4                            0                 100
```

Displays the details of the slice such as the slice ID, client rate, trunk rate, and the traffic percentage carried on the trunk ports. The **Encryption Supported** field indicates whether the slice is provisioned with firmware that supports encryption or not.

Note
The HW Status field might display "Need Upgrade" when the user needs to use the MACsec feature and upgrades from R6.0.1 to 6.1.1. Hence, the control FPGA (CTRL_BKP_UP, CTRL_BKP_LOW, CTRL_FPGA_UP, and CTRL_FPGA_LOW) needs to be upgraded to the latest firmware version provided by R6.1.1. See Verify Firmware Version for more information.

The Provisioned status does not indicate that the traffic can flow immediately. For example, use the **show controllers maCSecCtrlr 0/0/0/3** command output to view the provisioning information of the port after the slice is provisioned.

Example:

```
RP/0/RP0/CPU0:ios# show hw-module slice all


Thu Aug 11 16:16:58.935 IST
Slice ID:               0
Status:                 Provisioned
Client Bitrate:         100
Trunk Bitrate:          200
DP FPGA FW Type:        M100
```

```
DP FPGA FW Version:      02.00
HW Status:               CURRENT

Encryption Supported:    TRUE
Client Port -   Trunk Port          CoherentDSP0/0/0/6
Traffic Split Percentage

HundredGigECtrlr0/0/0/3                   100
HundredGigECtrlr0/0/0/4                   100

Slice ID:                1
Status:                  Provisioned
Client Bitrate:          100
Trunk Bitrate:           200
DP FPGA FW Type:         M100
DP FPGA FW Version:      02.00
HW Status:               CURRENT

Encryption Supported:    TRUE
Client Port -   Trunk Port          CoherentDSP0/0/0/13
Traffic Split Percentage

HundredGigECtrlr0/0/0/10                   100
HundredGigECtrlr0/0/0/11                   100

Slice ID:                2
Status:                  Provisioned
Client Bitrate:          100
Trunk Bitrate:           200
DP FPGA FW Type:         M100
DP FPGA FW Version:      02.00
HW Status:               CURRENT

Encryption Supported:    TRUE
Client Port -   Trunk Port          CoherentDSP0/0/0/20
Traffic Split Percentage

HundredGigECtrlr0/0/0/17                   100
HundredGigECtrlr0/0/0/18                   100

Slice ID:                3
Status:                  Provisioned
Client Bitrate:          100
Trunk Bitrate:           200
DP FPGA FW Type:         M100
DP FPGA FW Version:      02.00
HW Status:               CURRENT

Encryption Supported:    TRUE
Client Port -   Trunk Port          CoherentDSP0/0/0/27
Traffic Split Percentage

HundredGigECtrlr0/0/0/24                   100
HundredGigECtrlr0/0/0/25                   100
```

Associated Commands

- hw-module
- show hw-module

Apply MACsec Configuration on MACsec Controller

You can apply the MACsec key chain and policy configuration on the MACsec controller.

configure

controller MACSecCtrl *Rack/Slot/Instance/Port*

macsec psk-keychain *key-chain-name* [**policy** *policy-name*]

exit

commit

Example

```
configure
controller MACSecCtrl 0/0/0/3
macsec psk-keychain mac_chain policy mac_policy
exit
commit
```

Associated Commands

- controller mACSecCtrlr
- macsec psk-keychain

Verify MACsec Configuration on MACsec Controller

1. Verify the MACsec configuration on the controller.

   show macsec mka summary

   ```
   Wed Mar 30 13:35:15.497 UTC
   NODE: node0_RP0_CPU0
   ============================================================
    Interface      Status         Cipher-Suite        KeyChain
   ============================================================
    MS0/0/0/03      Secured        GCM-AES-XPN-256    mac_chain

   Total MACSec Sessions : 1
        Secured Sessions : 1
        Pending Sessions : 0
   ```

   The **Status** field in the output confirms that the respective controller is **Secured**. If MACsec encryption is not successfully configured, you will see a status such as **Pending** or **Init**.

2. Verify whether the MKA session is secured with MACsec on the respective controller.

   show macsec mka session

   ```
   Sun Dec 18 14:20:50.626 IST
   NODE: node0_RP0_CPU0
   ======================================================================
   ```

```
      Interface         Local-TxSCI       # Peers   Status  Key-Server
=========================================================================
     MS0/0/0/3      3820.563b.eacc/0003      1     Secured     YES
     MS0/0/0/18      3820.563b.eacc/0012     1     Secured     NO
     MS0/0/0/17      3820.563b.eacc/0011     1     Secured     NO
     MS0/0/0/4      3820.563b.eacc/0004      1     Secured     YES
```

show macsec mka session controller MS0/0/0/03 detail

```
Tue Aug 16 14:08:04.927 IST

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

Local Tx-SCI            : 3820.563b.eacc/0003
Local Tx-SSCI           : 2
Interface MAC Address   : 3820.563b.eacc
MKA Port Identifier     : 10
Interface Name          : MS0/0/0/03
CAK Name (CKN)         : ABC10000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)  : 6609E2B5F8ACC8653301503B
Message Number (MN)     : 39
Authenticator           : NO
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC

Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : 6609E2B5F8ACC8653301503B00000001 (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)

MKA Policy Name         : *DEFAULT POLICY*
Key Server Priority     : 16
Replay Window Size      : 64
Confidentiality Offset  : 0
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES

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

Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
  389DF014D752B8065B548283  62         3820.563b.eacc/0012     1     16

Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
```

The **Status** field in the output verifies if the MKA session is secured with MACsec encryption. The output also displays information about the controller and other MACsec parameters.

3. Verify the MACsec session counter statistics.

   show macsec mka statistics location 0/RP0/CPU0

```
Thu Aug 11 16:02:41.330 IST

MKA Global Statistics
=====================
MKA Session Totals
   Secured..................... 0
   Reauthentication Attempts.. 0

   Deleted (Secured).......... 0
   Keepalive Timeouts......... 0

CA Statistics
   Pairwise CAKs Derived...... 0
   Pairwise CAK Rekeys........ 0
   Group CAKs Generated....... 0
   Group CAKs Received........ 0

SA Statistics
   SAKs Generated............. 0
   SAKs Rekeyed............... 0
   SAKs Received.............. 0
   SAK Responses Received..... 0

MKPDU Statistics
   MKPDUs Validated & Rx...... 5
      "Distributed SAK"..... 0
      "Distributed CAK"..... 0
   MKPDUs Transmitted......... 4
      "Distributed SAK"..... 0
      "Distributed CAK"..... 0

MKA Error Counter Totals
========================
Session Failures
   Bring-up Failures................ 0
   Reauthentication Failures........ 0
   Duplicate Auth-Mgr Handle........ 0

SAK Failures
   SAK Generation................... 0
   Hash Key Generation.............. 0
   SAK Encryption/Wrap.............. 0
   SAK Decryption/Unwrap............ 0
   SAK Cipher Mismatch.............. 0

CA Failures
   Group CAK Generation............. 0
   Group CAK Encryption/Wrap........ 0
   Group CAK Decryption/Unwrap...... 0
   Pairwise CAK Derivation.......... 0
   CKN Derivation................... 0
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SC Creation................... 0
   Tx SC Creation................... 0
   Rx SA Installation............... 0
   Tx SA Installation............... 0

MKPDU Failures
   MKPDU Tx......................... 0
```

```
   MKPDU Rx Validation.............. 0
   MKPDU Rx Bad Peer MN............. 0
   MKPDU Rx Non-recent Peerlist MN.. 0
   MKPDU Rx Drop SAKUSE, KN mismatch...... 0
   MKPDU Rx Drop SAKUSE, Rx Not Set....... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.. 0
   MKPDU Rx Drop SAKUSE, AN Not in Use.... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set. 0

IOX Global Statistics
   MKPDUs Rx IDB not found.... 0
   MKPDUs Rx Invalid CKN...... 0
   MKPDUs Tx Invalid IDB...... 0
   MKPDUs Tx Pkt Build Fail... 0
```

The counters display the MACsec PDUs transmitted, validated, and received. The output also displays transmission errors, if any. This completes the verification of MACsec encryption on NCS 1002.

a. Verify the status of the MACsec controller.

show macsec platform status controller MacSecCtrlr 0/0/0/3

```
   Mon Jun  6 20:57:15.900 UTC

   -------------------------------------

   Interface Status

   -------------------------------------
       ReplayWindowSize       :  64
       MustSecure             :  TRUE
       SecureMode             :  2

   -------------------------------------

   Encrypted Secure Channel Status

   -------------------------------------
       ProtectionEnabled      :  TRUE
       SecureChannelID        :  0x0200d05a57395540
       ConfidentialityOffset  :  0
       CipherSuite            :  GCM-AES-XPN-256
       SecureTagLength        :  16
       InitialPacketNumber    :  1
       MaxPacketNumber        :  18446744073709551615
       RecentPacketNumber     :  364865080

   -------------------------------------

   Encrypted Active Associations

   -------------------------------------
       AssociationNumber      :  1
       DeviceAssociationNum   :  1
       ShortSecureChannelID   :  1
       ProgrammedTime         :  2016 Jun 6 20:57:09.690
       KeyCRC                 :  0x6fe6f59c
       XpnSalt                :  0xffca89c5 0x4a307f93 0xd3df482e


   -------------------------------------

   Decrypted Secure Channel Status
```

```
-------------------------------------
    ProtectionEnabled      :  TRUE
    SecureChannelID        :  0x0100d05a57395540
    ConfidentialityOffset  :  0
    CipherSuite            :  GCM-AES-XPN-256
    InitialPacketNumber    :  1
    MaxPacketNumber        :  18446744073709551615
    RecentPacketNumber     :  370010268

-------------------------------------

Decrypted Active Associations

-------------------------------------
    AssociationNumber      :  1
    DeviceAssociationNum   :  1
    ShortSecureChannelID   :  2
    ProgrammedTime         :  2016 Jun 6 20:57:09.550
    KeyCRC                 :  0x6fe6f59c
    XpnSalt                :  0xfcca89c5 0x4a307f93 0xd3df482e
```

When IOS XR is reloaded, two association numbers are displayed under Decrypted Active Associations. After the reload, key roll over is required. When the key rollover happens, the active association number is associated.

Verify State of MACSec Controller

The state of MACSec controller can be verified using the **show controllers MACSecCtrlr** *R/S/I/P* command. If the state of MACSec controller is down, the corresponding MKA sessions do not come up.

The state of MACSec controller is down upon one of the following conditions.

- State of the corresponding Ethernet controller is Admin Down. The state can be verified using the **show controllers HundredGigECtrlr** *R/S/I/P* command.
- State of the optics controller is Admin Down or Operational Down. The state can be verified using the **show controllers optics** *R/S/I/P* command.
- Client optics is not present. The client optics can be verified using the **show inventory** command.

The state of the Ethernet controller can be changed from Admin Down using the following commands.

```
configure
controller HundredGigECtrlr Rack/Slot/Instance/Port
no shutdown
commit
```

The state of the optics controller can be changed from Admin Down or Operational Down using the following commands.

```
configure
controller optics Rack/Slot/Instance/Port
no shutdown
commit
```

SecY Statistics

SecY statistics is used to identify issues with the encrypted traffic.

Before You Begin

Ensure that MKA sessions are established. See Verify MACsec Configuration on MACsec Controller, on page 19 for more information.

100G MACsec

show macsec secy stats controller MACSecCtrlr 0/0/0/3 SC

```
Tue Jan 22 04:42:32.044 IST
Interface Stats
    InPktsUntagged     :  0
    InPktsNoTag        :  0
    InPktsBadTag       :  0
    InPktsUnknownSCI   :  0
    InPktsNoSCI        :  0
    InPktsOverrun      :  0
    InOctetsValidated  :  0
    InOctetsDecrypted  :  121697919056
    OutPktsUntagged    :  0
    OutPktsTooLong     :  0
    OutOctetsProtected :  0
    OutOctetsEncrypted :  194316914428

SC Stats
  TxSC Stats
    OutPktsProtected   : 0
    OutPktsEncrypted   : 130941317
    OutOctetsProtected : 0
    OutOctetsEncrypted : 194316914428
    OutPktsTooLong     : 0
    TxSA Stats
      TxSA 0:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0
      TxSA 1:
        OutPktsProtected : 0
        OutPktsEncrypted : 130941317
        NextPN           : 130940105
      TxSA 2:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0
      TxSA 3:
        OutPktsProtected : 0
        OutPktsEncrypted : 0
        NextPN           : 0

  RxSC Stats
    RxSC 1: 0
      InPktsUnchecked     : 0
      InPktsDelayed       : 0
      InPktsLate          : 0
      InPktsOK            : 82006684
      InPktsInvalid       : 0
      InPktsNotValid      : 0
```

```
      InPktsNotUsingSA    : 0
      InPktsUnusedSA      : 0
      InPktsUntaggedHit   : 0
      InOctetsValidated   : 0
      InOctetsDecrypted   : 121697919056
    RxSA Stats
      RxSA 0:
        InPktsUnusedSA      : 0
        InPktsNotUsingSA    : 0
        InPktsNotValid      : 0
        InPktsInvalid       : 0
        InPktsOK            : 0
        NextPN              : 1
      RxSA 1:
        InPktsUnusedSA      : 0
        InPktsNotUsingSA    : 0
        InPktsNotValid      : 0
        InPktsInvalid       : 0
        InPktsOK            : 82006684
        NextPN              : 82004142
      RxSA 2:
        InPktsUnusedSA      : 0
        InPktsNotUsingSA    : 0
        InPktsNotValid      : 0
        InPktsInvalid       : 0
        InPktsOK            : 0
        NextPN              : 0
      RxSA 3:
        InPktsUnusedSA      : 0
        InPktsNotUsingSA    : 0
        InPktsNotValid      : 0
        InPktsInvalid       : 0
        InPktsOK            : 0
        NextPN              : 0
```

The SecY SA counters are displayed as 64 bit values in the CLI.

10G MACsec

show macsec secy stats controller MACSecCtrlr 0/0/0/3/1 SC

```
Mon Dec 19 17:04:00.467 IST
Interface Stats
    InPktsUntagged     :  0
    InPktsNoTag        :  0
    InPktsBadTag       :  0
    InPktsUnknownSCI   :  0
    InPktsNoSCI        :  0
    InPktsOverrun      :  0
    InOctetsValidated  :  0
    InOctetsDecrypted  :  3244694362816
    OutPktsUntagged    :  0
    OutPktsTooLong     :  0
    OutOctetsProtected :  0
    OutOctetsEncrypted :  3225943872072

SC Stats
  TxSC Stats
    OutPktsProtected   : 0
    OutPktsEncrypted   : 336597056
    OutOctetsProtected : 0
    OutOctetsEncrypted : 3225943872072
    OutPktsTooLong     : 0

 RxSC Stats
```

```
RxSC 1: 0
  InPktsUnchecked     : 0
  InPktsDelayed       : 0
  InPktsLate          : 0
  InPktsOK            : 338553493
  InPktsInvalid       : 0
  InPktsNotValid      : 0
  InPktsNotUsingSA    : 1320396
  InPktsUnusedSA      : 0
  InPktsUntaggedHit   : 0
  InOctetsValidated   : 0
  InOctetsDecrypted   : 3244694362816
```

Trunk Side Statistics

Trunk side statistics is used to isolate issues with the encrypt and decrypt blocks. In the Tx direction, the trunk side Egress statistics display statistics after the encrypt block. In the Rx direction, the trunk side Ingress statistics display statistics before the decrypt block.

show controllers MACSecCtrlr 0/0/0/3 stats

```
Tue Jan 22 04:51:40.858 IST
Statistics for interface MACSecCtrlr0/0/0/3 (cached values):

Ingress:
    Input total bytes           = 805443936740
    Input good bytes            = 805443936740

    Input total packets         = 525746695
    Input 802.1Q frames         = 0
    Input pause frames          = 0
    Input pkts 64 bytes         = 0
    Input pkts 65-127 bytes     = 0
    Input pkts 128-255 bytes    = 0
    Input pkts 256-511 bytes    = 0
    Input pkts 512-1023 bytes   = 0
    Input pkts 1024-1518 bytes  = 0
    Input pkts 1519-Max bytes   = 0

    Input good pkts             = 525746695
    Input unicast pkts          = 0
    Input multicast pkts        = 0
    Input broadcast pkts        = 0

    Input drop overrun          = 0
    Input drop abort            = 0
    Input drop invalid VLAN     = 0
    Input drop invalid DMAC     = 0
    Input drop invalid encap    = 0
    Input drop other            = 0

    Input error giant           = 0
    Input error runt            = 0
    Input error jabbers         = 0
    Input error fragments       = 0
    Input error CRC             = 0
    Input error collisions      = 0
    Input error symbol          = 0
    Input error other           = 0
```

```
    Input MIB giant             = 0
    Input MIB jabber            = 0
    Input MIB CRC               = 0

Egress:
    Output total bytes          = 880411742408
    Output good bytes           = 880411742408

    Output total packets        = 574681294
    Output 802.1Q frames        = 0
    Output pause frames         = 0
    Output pkts 64 bytes        = 0
    Output pkts 65-127 bytes    = 0
    Output pkts 128-255 bytes   = 0
    Output pkts 256-511 bytes   = 0
    Output pkts 512-1023 bytes  = 0
    Output pkts 1024-1518 bytes = 0
    Output pkts 1519-Max bytes  = 0

    Output good pkts            = 574681294
    Output unicast pkts         = 0
    Output multicast pkts       = 0
    Output broadcast pkts       = 0

    Output drop underrun        = 0
    Output drop abort           = 0
    Output drop other           = 0

    Output error other          = 0
```

clear controller MACSecCtrlr 0/0/0/3 stats

```
Tue Jan 22 04:52:40.858 IST
```

show controllers macSecCtrlr 0/0/0/3 stats | inc total

```
Tue Jan 22 04:51:45.227 IST
    Input total bytes           = 805443936740
    Input total packets         = 525746695
    Output total bytes          = 880411742408
    Output total packets        = 574681294
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/3 stats | inc total
Tue Jan 22 04:51:47.695 IST
    Input total bytes           = 805443936740
    Input total packets         = 525746695
```

Control Plane Statistics

show macsec mka statistics controller macSecCtrlr 0/0/0/3

This command displays control plane statistics for the specific MACSec controller.

```
Tue Jan 22 04:57:57.450 IST


MKA Statistics for Session on interface (MS0/0/0/3)
=======================================================
Reauthentication Attempts.. 0

CA Statistics
   Pairwise CAKs Derived... 0
```

```
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0

SA Statistics
   SAKs Generated.......... 1
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 1

MKPDU Statistics
   MKPDUs Transmitted...... 3305
      "Distributed SAK".. 1
      "Distributed CAK".. 0
   MKPDUs Validated & Rx... 3305
      "Distributed SAK".. 0
      "Distributed CAK".. 0

MKA IDB Statistics
   MKPDUs Tx Success.......... 3305
   MKPDUs Tx Fail............. 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUS No Tx on intf down.. 2
   MKPDUS No Rx on intf down.. 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error............ 0
   MKPDUs Rx Success.......... 3305
   MKPDUs Rx Invalid Length... 0
   MKPDUs Rx Invalid CKN...... 0

MKPDU Failures
   MKPDU Rx Validation (ICV)................. 0
   MKPDU Rx Bad Peer MN...................... 0
   MKPDU Rx Non-recent Peerlist MN........... 0
   MKPDU Rx Drop SAKUSE, KN mismatch......... 0
   MKPDU Rx Drop SAKUSE, Rx Not Set.......... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch..... 0
   MKPDU Rx Drop SAKUSE, AN Not in Use....... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set.... 0
   MKPDU Rx Drop Packet, Ethertype Mismatch.. 0
   MKPDU Rx Drop Packet, Source MAC NULL..... 0
   MKPDU Rx Drop Packet, Destination MAC NULL 0
   MKPDU Rx Drop Packet, Payload NULL........ 0

SAK Failures
   SAK Generation.................. 0
   Hash Key Generation............. 0
   SAK Encryption/Wrap............. 0
   SAK Decryption/Unwrap........... 0

CA Failures
   ICK Derivation.................. 0
   KEK Derivation.................. 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SC Creation.................. 0
   Tx SC Creation.................. 0
   Rx SA Installation.............. 0
   Tx SA Installation.............. 0
```

clear macsec mka statistics controller macSecCtrlr 0/0/0/3

This command clears control plane statistics for the specific MACSec controller.

```
Tue Jan 22 04:59:33.830 IST
```

Configuring MACsec Threshold Crossing Alerts

You can configure MACsec Threshold Crossing Alerts (TCA) at mac-sec ether, secy-if (interface), and secy-tx. There is no default threshold, mimimum or maximum threshold for configuring MACsec TCA. You can configure it between the range 1 to 4294967295. By default, you can find TCA in **show logging** command. You can configure syslog server in MACsec TCA, and view TCA in syslog server such as EPNM speicific parameter. You can set TCA for all supported buckets such as 0-sec/15-mins/24-hour for parameters in mac-sec ether/ secy-if/ secy-tx, secy-rx.

Use the following command to configure the MACsec Threshold Crossing Alerts (TCA) at MACsec ether layer, MACsec-secy-if, MACsec-secy-Tx, and /or MACsec-secy-Rx:

controllers macSecCtrlr *R/S/P* {**pm** {**30 sec** | **15-min** | **24-hour**} {**macsec-ether** | **macsec-secy-if** | **macsec-secy-tx** | **macsec-secy-rx**} {**report | threshold** {**in-oct-decrypted | out-oct-decrypted**} *value*} **enable**

Examples

The following is a sample to configure the MACsec TCA parameters for rx-pkt at macsec-ether level for MACsec controller in 15 min intervals:

```
controllers macSecCtrlr 0/0/0/11/1
 pm 15-min macsec-ether report rx-pkt enable
pm 15-min macsec-ether threshold rx-pkt 1000000
```

The following is a sample to configure the MACsec TCA parameters for rx-util at macsec-ether level for MACsec controller in 15 min intervals:

```
controllers macSecCtrlr 0/0/0/11/1
pm 15-min macsec-ether report rx-util enable
pm 15-min macsec-ether threshold rx-util 10
```

The following is a sample to configure the MACsec TCA parameters for out-octets at macsec-ether level for MACsec controller in 15 min intervals:

```
controllers macSecCtrlr 0/0/0/11/1
pm 15-min macsec-ether report out-octets enable
pm 15-min macsec-ether threshold out-octets 100000
```

The following is a sample to configure the MACsec TCA parameters for rx-pkt at MAC-SECy-If controller in 30 sec interval:

```
controller MACSecCtrlr0/0/0/4
pm 15-min macsec-ether report rx-pkt enable
pm 15-min macsec-ether threshold rx-pkt 1000
```

Clear Commands

You can use the following commands to clear the PMs across different buckets:

```
clear controller macSecCtrlr 0/0/0/10 pm 15-min
clear controller macSecCtrlr 0/0/0/10 pm 30-sec
clear controller macSecCtrlr 0/0/0/10 pm 24-hour
```

View MACsec PM Parameters

You must configure MACSec controllers to view MACsec performance. To configure MACSec controllers , seeApply MACsec Configuration on MACsec Controller, on page 19.

Use the following commands to view the MACsec performance monitoring at MACsec ether layer, MACsec-secy-if, MACsec-secy-Tx, and /or MACsec-secy-Rx:

show controllers macSecCtrlr *R/S/P*{ **pm**{**current** |**history**} { **30 sec** |**15-min** | **24-hour** } {**macsec-ether** | **macsec-secy-if** | **macsec-secy-tx** | **macsec-secy-rx**}

```
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/9/1 pm current 30-sec macsec-ether
```

Displays the current performance monitoring parameters of the Ethernet controller in 30 second interval.

```
ETHER in the current interval [23:10:30 - 23:10:31 Sat Mar 18 2017]

ETHER current bucket type : Valid
 RX-UTIL[%]                : 92.75      Threshold : 0.00      TCA(enable) : NO
 TX-UTIL[%]                : 82.26      Threshold : 0.00      TCA(enable) : NO
 RX-PKT                    : 1077504     Threshold : 0         TCA(enable) : NO
 STAT-PKT                  : 0           Threshold : 0         TCA(enable) : NO
 OCTET-STAT                : 1137844152  Threshold : 0         TCA(enable) : NO
 OVERSIZE-PKT              : 0           Threshold : 0         TCA(enable) : NO
 FCS-ERR                   : 0           Threshold : 0          TCA(enable) : NO
 LONG-FRAME                : 0           Threshold : 0          TCA(enable) : NO
 JABBER-STATS              : 0           Threshold : 0          TCA(enable) : NO
 64-OCTET                  : 0           Threshold : 0          TCA(enable) : NO
 65-127-OCTET              : 0           Threshold : 0           TCA(enable) : NO
 128-255-OCTET             : 0           Threshold : 0           TCA(enable) : NO
 256-511-OCTET             : 0           Threshold : 0           TCA(enable) : NO
 512-1023-OCTET            : 0           Threshold : 0           TCA(enable) : NO
 1024-1518-OCTET           : 0           Threshold : 0           TCA(enable) : NO
 IN-UCAST                  : 0           Threshold : 0           TCA(enable) : NO
 IN-MCAST                  : 0           Threshold : 0           TCA(enable) : NO
 IN-BCAST                  : 0           Threshold : 0           TCA(enable) : NO
 OUT-UCAST                 : 0           Threshold : 0           TCA(enable) : NO
 OUT-BCAST                 : 0            Threshold : 0          TCA(enable) : NO
 OUT-MCAST                 : 0            Threshold : 0          TCA(enable) : NO
 TX-PKT                    : 955636       Threshold : 0          TCA(enable) : NO
 OUT-OCTET                 : 1009150716   Threshold : 2000       TCA(enable) : NO
 IFIN-ERRORS               : 0            Threshold : 0          TCA(enable) : NO
 IFIN-OCTETS               : 0            Threshold : 0          TCA(enable) : NO
 STAT-MULTICAST-PKT        : 0            Threshold : 0          TCA(enable) : NO
 STAT-BROADCAST-PKT        : 0            Threshold : 0          TCA(enable) : NO
 STAT-UNDERSIZED-PKT       : 0            Threshold : 0          TCA(enable) : NO
 IN_GOOD_BYTES             : 0            Threshold : 0          TCA(enable) : NO
 IN_GOOD_PKTS              : 0            Threshold : 0          TCA(enable) : NO
 IN_DROP_OTHER             : 0            Threshold : 0          TCA(enable) : NO
 IN_ERROR_FRAGMENTS        : 0            Threshold : 0          TCA(enable) : NO
 IN_PKT_64_OCTET           : 0            Threshold : 0          TCA(enable) : NO
 IN_PKTS_65_127_OCTETS     : 0            Threshold : 0          TCA(enable) : NO
 IN_PKTS_128_255_OCTETS    : 0            Threshold : 0          TCA(enable) : NO
 IN_PKTS_256_511_OCTETS    : 0            Threshold : 0          TCA(enable) : NO
 IN_PKTS_512_1023_OCTETS   : 0            Threshold : 0          TCA(enable) : NO
 IN_PKTS_1024_1518_OCTETS  : 0            Threshold : 0          TCA(enable) : NO
 TX_UNDERSIZED_PKT         : 0            Threshold : 0          TCA(enable) : NO
 TX_OVERSIZED_PKT          : 0            Threshold : 0          TCA(enable) : NO
 TX_FRAGMENTS              : 0            Threshold : 0          TCA(enable) : NO
 TX_JABBER                 : 0            Threshold : 0          TCA(enable) : NO
 TX_BAD_FCS                : 0            Threshold : 0          TCA(enable) : NO
```

Example

```
ios#show controllers macSecCtrlr 0/0/0/9/1 pm current 15-min macsec-ether
```

Displays the current performance monitoring parameters of the Ethernet controller in 15 minute intervals

```
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/9/1 pm current 15-min macsec-ether
Sat Mar 18 23:10:41.410 IST
ETHER in the current interval [23:00:00 - 23:10:41 Sat Mar 18 2017]
ETHER current bucket type : Valid
 RX-UTIL[%]                : 92.75      Threshold : 0.00      TCA(enable) : NO
 TX-UTIL[%]                : 82.26      Threshold : 0.00      TCA(enable) : NO
 RX-PKT                    : 690733237            Threshold : 0            TCA(enable) :
NO
 STAT-PKT                  : 0                    Threshold : 0            TCA(enable) :
NO
 OCTET-STAT                : 729414298536         Threshold : 0            TCA(enable) :
NO
 OVERSIZE-PKT              : 0                    Threshold : 0            TCA(enable) :
NO
 FCS-ERR                   : 0                    Threshold : 0            TCA(enable) :
NO
 LONG-FRAME                : 0                    Threshold : 0            TCA(enable) :
NO
 JABBER-STATS              : 0                    Threshold : 0            TCA(enable) :
NO
 64-OCTET                  : 0                    Threshold : 0            TCA(enable) :
NO
 65-127-OCTET              : 0                    Threshold : 0            TCA(enable) :
NO
 128-255-OCTET             : 0                    Threshold : 0            TCA(enable) :
NO
 256-511-OCTET             : 0                    Threshold : 0            TCA(enable) :
NO
 512-1023-OCTET            : 0                    Threshold : 0            TCA(enable) :
NO
 1024-1518-OCTET           : 0                    Threshold : 0            TCA(enable) :
NO
 IN-UCAST                  : 0                    Threshold : 0            TCA(enable) :
NO
 IN-MCAST                  : 0                    Threshold : 0            TCA(enable) :
NO
 IN-BCAST                  : 0                    Threshold : 0            TCA(enable) :
NO
 OUT-UCAST                 : 0                    Threshold : 0            TCA(enable) :
NO
 OUT-BCAST                 : 0                    Threshold : 0            TCA(enable) :
NO
 OUT-MCAST                 : 0                    Threshold : 0            TCA(enable) :
NO
 TX-PKT                    : 612577128            Threshold : 0            TCA(enable) :
NO
 OUT-OCTET                 : 646881447168         Threshold : 8000         TCA(enable) :
NO
 IFIN-ERRORS               : 0                    Threshold : 0            TCA(enable) :
NO
 IFIN-OCTETS               : 0                    Threshold : 0            TCA(enable) :
NO
 STAT-MULTICAST-PKT        : 0                    Threshold : 0            TCA(enable) :
NO
 STAT-BROADCAST-PKT        : 0                    Threshold : 0            TCA(enable) :
NO
 STAT-UNDERSIZED-PKT       : 0                    Threshold : 0            TCA(enable) :
NO
 IN_GOOD_BYTES             : 0                    Threshold : 0            TCA(enable) :
```

```
NO
 IN_GOOD_PKTS              : 0                    Threshold : 0            TCA(enable) :
NO
 IN_DROP_OTHER             : 0                    Threshold : 0            TCA(enable) :
NO
 IN_ERROR_FRAGMENTS        : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKT_64_OCTET           : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKTS_65_127_OCTETS     : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKTS_128_255_OCTETS    : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKTS_256_511_OCTETS    : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKTS_512_1023_OCTETS   : 0                    Threshold : 0            TCA(enable) :
NO
 IN_PKTS_1024_1518_OCTETS  : 0                    Threshold : 0            TCA(enable) :
NO
 TX_UNDERSIZED_PKT         : 0                    Threshold : 0            TCA(enable) :
NO
 TX_OVERSIZED_PKT          : 0                    Threshold : 0            TCA(enable) :
NO
 TX_FRAGMENTS              : 0                    Threshold : 0            TCA(enable) :
NO
 TX_JABBER                 : 0                    Threshold : 0            TCA(enable) :
NO
 TX_BAD_FCS                : 0                    Threshold : 0            TCA(enable) :
NO
```

Example

```
ios#show controllers macSecCtrlr 0/0/0/9/1 pm current 24-hour macsec-ether
```

Displays the current performance monitoring parameters of the Ethernet controller in 24 hour intervals.

```
ios#show controllers macSecCtrlr 0/0/0/9/1 pm current 24-hour macsec-ether
Sat Mar 18 23:10:49.939 IST
ETHER in the current interval [00:00:00 - 23:10:50 Sat Mar 18 2017]
ETHER current bucket type : Invalid
 RX-UTIL[%]                : 92.75      Threshold : 0.00      TCA(enable) : NO
 TX-UTIL[%]                : 82.26      Threshold : 0.00      TCA(enable) : NO
 RX-PKT                    : 1508621424           Threshold : 0            TCA(enable) :
NO
 STAT-PKT                  : 0                    Threshold : 0            TCA(enable) :
NO
 OCTET-STAT                : 1593104223188        Threshold : 0            TCA(enable) :
NO
 OVERSIZE-PKT              : 0                    Threshold : 0            TCA(enable) :
NO
 FCS-ERR                   : 0                    Threshold : 0            TCA(enable) :
NO
 LONG-FRAME                : 0                    Threshold : 0            TCA(enable) :
NO
 JABBER-STATS              : 0                    Threshold : 0            TCA(enable) :
NO
 64-OCTET                  : 0                    Threshold : 0            TCA(enable) :
NO
 65-127-OCTET              : 0                    Threshold : 0            TCA(enable) :
NO
 128-255-OCTET             : 0                    Threshold : 0            TCA(enable) :
NO
 256-511-OCTET             : 0                    Threshold : 0            TCA(enable) :
NO
 512-1023-OCTET            : 0                    Threshold : 0            TCA(enable) :
```

```
NO
 1024-1518-OCTET           : 0                    Threshold : 0           TCA(enable) :
NO
 IN-UCAST                  : 0                    Threshold : 0           TCA(enable) :
NO
 IN-MCAST                  : 0                    Threshold : 0           TCA(enable) :
NO
 IN-BCAST                  : 0                    Threshold : 0           TCA(enable) :
NO
 OUT-UCAST                 : 0                    Threshold : 0           TCA(enable) :
NO
 OUT-BCAST                 : 0                    Threshold : 0           TCA(enable) :
NO
 OUT-MCAST                 : 0                    Threshold : 0           TCA(enable) :
NO
 TX-PKT                    : 1337921648           Threshold : 0           TCA(enable) :
NO
 OUT-OCTET                 : 1412845260584        Threshold : 100000      TCA(enable) :
NO
 IFIN-ERRORS               : 0                    Threshold : 0           TCA(enable) :
NO
 IFIN-OCTETS               : 0                    Threshold : 0           TCA(enable) :
NO
 STAT-MULTICAST-PKT        : 0                    Threshold : 0           TCA(enable) :
NO
 STAT-BROADCAST-PKT        : 0                    Threshold : 0           TCA(enable) :
NO
 STAT-UNDERSIZED-PKT       : 0                    Threshold : 0           TCA(enable) :
NO
 IN_GOOD_BYTES             : 0                    Threshold : 0           TCA(enable) :
NO
 IN_GOOD_PKTS              : 0                    Threshold : 0           TCA(enable) :
NO
 IN_DROP_OTHER             : 0                    Threshold : 0           TCA(enable) :
NO
 IN_ERROR_FRAGMENTS        : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKT_64_OCTET           : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKTS_65_127_OCTETS     : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKTS_128_255_OCTETS    : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKTS_256_511_OCTETS    : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKTS_512_1023_OCTETS   : 0                    Threshold : 0           TCA(enable) :
NO
 IN_PKTS_1024_1518_OCTETS  : 0                    Threshold : 0           TCA(enable) :
NO
 TX_UNDERSIZED_PKT         : 0                    Threshold : 0           TCA(enable) :
NO
 TX_OVERSIZED_PKT          : 0                    Threshold : 0           TCA(enable) :
NO
 TX_FRAGMENTS              : 0                    Threshold : 0           TCA(enable) :
NO
 TX_JABBER                 : 0                    Threshold : 0           TCA(enable) :
NO
 TX_BAD_FCS                : 0                    Threshold : 0           TCA(enable) :
NO
```

Example

```
ios#show controllers macSecCtrlr 0/0/0/16/2 pm current 30-sec macsec-secy-if
```

Displays the current performance monitoring parameters of the controller in macsec-secy-if mode in 30 sec intervals.

```
Macsec-Secy-If in the current interval [10:18:30 - 10:18:57 Sat Apr 22 2017]
Macsec-Secy-If current bucket type : Valid
InPktsUntagged     : 0                    Threshold : 0          TCA(enable) : NO
InPktsNoTag        : 0                    Threshold : 0          TCA(enable) : NO
InPktsBadTag       : 0                    Threshold : 0          TCA(enable) : NO
InPktsUnknownSCI   : 0                    Threshold : 0          TCA(enable) : NO
InPktsNoSCI        : 0                    Threshold : 0          TCA(enable) : NO
InPktsOverrun      : 0                    Threshold : 0          TCA(enable) : NO
InOctetsValidated  : 0                    Threshold : 0          TCA(enable) : NO
InOctetsDecrypted  : 321909392            Threshold : 0          TCA(enable) : NO
OutPktsUntagged    : 0                    Threshold : 0          TCA(enable) : NO
OutPktsTooLong     : 0                    Threshold : 0          TCA(enable) : NO
OutOctetsProtected : 0                    Threshold : 0          TCA(enable) : NO
OutOctetsEncrypted : 415501264            Threshold : 0          TCA(enable) : NO
```

Example

```
ios#show controllers macSecCtrlr 0/0/0/16/2 pm current 15-min macsec-secy-if
```

Displays the current performance monitoring parameters of the controller in macsec-secy-if mode in 15 minute intervals.

```
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/16/2 pm current 15-min macsec-secy-if
Sat Apr 22 10:18:40.743 UTC
Macsec-Secy-If in the current interval [10:15:00 - 10:18:40 Sat Apr 22 2017]
Macsec-Secy-If current bucket type : Valid
 InPktsUntagged     : 0                    Threshold : 0          TCA(enable) : NO
 InPktsNoTag        : 0                    Threshold : 0          TCA(enable) : NO
 InPktsBadTag       : 0                    Threshold : 0          TCA(enable) : NO
 InPktsUnknownSCI   : 0                    Threshold : 0          TCA(enable) : NO
 InPktsNoSCI        : 0                    Threshold : 0          TCA(enable) : NO
 InPktsOverrun      : 0                    Threshold : 0          TCA(enable) : NO
 InOctetsValidated  : 0                    Threshold : 0          TCA(enable) : NO
 InOctetsDecrypted  : 2541082096           Threshold : 0          TCA(enable) : NO
 OutPktsUntagged    : 0                    Threshold : 0          TCA(enable) : NO
 OutPktsTooLong     : 0                    Threshold : 0          TCA(enable) : NO
 OutOctetsProtected : 0                    Threshold : 0          TCA(enable) : NO
 OutOctetsEncrypted : 3279875344           Threshold : 0          TCA(enable) : NO
```

Example

```
ios#show controllers macSecCtrlr 0/0/0/16/2 pm current 30-sec  macsec-secy-tx
```

Displays the current performance monitoring parameters of the controller in macsec-secy-tx mode in 30 minute intervals.

```
Macsec-Secy-Tx in the current interval [10:18:30 - 10:18:59 Sat Apr 22 2017]
Macsec-Secy-Tx current bucket type : Valid
OutPktsProtected   : 0                    Threshold : 0          TCA(enable) : NO
OutPktsEncrypted   : 286527               Threshold : 0          TCA(enable) : NO
OutOctetsProtected : 0                    Threshold : 0          TCA(enable) : NO
OutOctetsEncrypted : 430363554            Threshold : 0          TCA(enable) : NO
OutPktsTooLong     : 0                    Threshold : 0          TCA(enable) : NO
```

Displays the current performance monitoring parameters of the controller in macsec-secy-tx mode in 24-hour interval.

```
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/24/3 pm current 24-hour macsec-secy-tx
Sat Apr  1 15:38:30.158 IST
Macsec-Secy-Tx in the current interval [00:00:00 - 15:38:30 Sat Apr 1 2017]
Macsec-Secy-Tx current bucket type : Valid
OutPktsProtected   : 0                    Threshold : 0          TCA(enable) : NO
```

```
OutPktsEncrypted   : 3160983513          Threshold : 0          TCA(enable) : NO
OutOctetsProtected : 0                   Threshold : 0          TCA(enable) : NO
OutOctetsEncrypted : 31559259393792      Threshold : 0          TCA(enable) : NO
OutPktsTooLong     : 0                   Threshold : 0          TCA(enable) : NO
```

Displays the current performance monitoring parameters of the controller in macsec-secy-rx mode in 24-hour interval.

```
RP/0/RP0/CPU0:ios#show controllers macSecCtrlr 0/0/0/10/3 pm current 24-hour macsec-secy-rx
Sat Apr  1 15:38:00.820 IST
Macsec-Secy-Rx in the current interval [00:00:00 - 15:38:01 Sat Apr 1 2017]
Macsec-Secy-Rx current bucket type : Valid
InPktsUnchecked   : 0                    Threshold : 0          TCA(enable) : NO
InPktsDelayed     : 0                    Threshold : 0          TCA(enable) : NO
InPktsLate        : 0                    Threshold : 0          TCA(enable) : NO
InPktsInvalid     : 0                    Threshold : 0          TCA(enable) : NO
InPktsOK          : 3159299558           Threshold : 0          TCA(enable) : NO
InPktsNotValid    : 0                    Threshold : 0          TCA(enable) : NO
InPktsNotUsingSA  : 0                    Threshold : 0          TCA(enable) : NO
InPktsUnusedSA    : 0                    Threshold : 0          TCA(enable) : NO
InPktsUntaggedHit : 0                    Threshold : 0          TCA(enable) : NO
InOctetsValidated : 0                    Threshold : 0          TCA(enable) : NO
InOctetsDecrypted : 31542446787072       Threshold : 0          TCA(enable) : NO
```

MACsec MKA Using EAP-TLS Authentication

Using IEEE 802.1X port-based authentication with Extensible Authentication Protocol (EAP-TLS), MACsec MKA can be configured between two NCS 1002 device ports. EAP-TLS allows mutual authentication and obtains MSK (master session key or primary session key). Both Connectivity Association Key Name (CKN) and connectivity association key (CAK) are derived from MSK for MKA operations. The device certificates are carried for authentication to the external AAA server using EAP-TLS.

IEEE 802.1X Device Roles

The devices in the network have the following specific roles with IEEE 802.1X authentication.

- Supplicant - An entity at one end of a point-to-point LAN segment that seeks to be authenticated by an Authenticator attached to the other end of that link.
- Authenticator - An entity that facilitates authentication of other entities attached to the same LAN.
- Authentication Server - An entity that provides an authentication service to an Authenticator. This service determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the system in which the Authenticator resides.

Prerequisites for MACsec MKA Using EAP-TLS Authentication

- Ensure that a Certificate Authority (CA) server is configured for the network.
- Ensure a valid CA certificate.
- Ensure that the user has configured Cisco Identity Services Engine (ISE) Release 2.2 onwards or Cisco Secure Access Control Server Release 5.6 onwards as external AAA server.

- It is always good to have the NCS 1002 devices, the CA server, and the external AAA/Radius server synchronized using Network Time Protocol (NTP) Server. If clock is not synchronized, there might be instances where certificate validation will not be successful, due to timing issues.

  However, there is no dependency on the timezone between NCS 1002 devices, the CA Server and the external AAA/Radius server.

Configure MACsec Encryption Using EAP-TLS Authentication

Configuring MACsec encryption using EAP-TLS authentication involves the following tasks:

Configure RADIUS Server

configure

radius-server host {*IPv4 address of RADIUS server*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**key** *string*]

radius-server vsa attribute ignore unknown

exit

commit

Examples

The following is a sample of configuring the RADIUS server.

```
configure
radius-server host 209.165.200.225 auth-port 1645 acct-port 1646 key cisco
radius-server vsa attribute ignore unknown
exit
commit
```

The following is sample output of **show radius** command.

```
Tue Jun 27 10:39:20.851 IST
Global dead time: 0 minute(s)
```

```
Number of Servers: 1

Server: 209.165.200.225  is UP
  Address family: IPv4
  Total Deadtime: 0s Last Deadtime: 0s
  Timeout: 5 sec, Retransmit limit: 3
  Quarantined: No
  Authentication:
    42 requests, 0 pending, 0 retransmits
    6 accepts, 0 rejects, 0 challenges
    0 timeouts, 0 bad responses, 0 bad authenticators
    0 unknown types, 0 dropped, 361 ms latest rtt
    Throttled: 0 transactions, 0 timeout, 0 failures
    Estimated Throttled Access Transactions: 0
    Maximum Throttled Access Transactions: 0

    Automated TEST Stats:
        0 requests, 0 timeouts, 0 response, 0 pending
  Accounting:
    0 requests, 0 pending, 0 retransmits
    0 responses, 0 timeouts, 0 bad responses
    0 bad authenticators, 0 unknown types, 0 dropped
    0 ms latest rtt
    Throttled: 0 transactions, 0 timeout, 0 failures
    Estimated Throttled Accounting Transactions: 0
    Maximum Throttled Accounting Transactions: 0

    Automated TEST Stats:
        0 requests, 0 timeouts, 0 response, 0 pending
```

Associated Commands

- radius-server
- radius-server host

Configure 802.1X Authentication Method

This procedure allows the user to configure 802.1X authentication method using RADIUS as the protocol. 802.1X authentication configuration allows to configure non-default profiles. However, only default is supported in NCS 1002.

configure

aaa authentication dot1x default group radius

exit

commit

Examples

The following is a sample of configuring the 802.1X authentication method.

```
configure
aaa authentication dot1x default group radius
exit
commit
```

The following is sample output of **show run aaa** command.

```
Tue Jun 27 10:39:17.437 IST
radius-server vsa attribute ignore unknown
radius-server host 209.165.200.225 auth-port 1645 acct-port 1646 key cisco
aaa authentication dot1x default group radius
```

Associated Commands

- aaa authentication dot1x

Generate RSA Key Pair

RSA key pairs are used to sign and encrypt key management messages. This is required before the you can obtain a certificate for the node.

configure

crypto key generate rsa [**usage-keys** | **general-keys**] [*keypair-label*]

exit

commit

Examples

The following is a sample of generating the RSA key pair.

```
configure
crypto key generate rsa ncs1k
exit
commit
```

The following is sample output of **show crypto key** *key-name* **rsa** command.

```
Tue Jun 27 10:39:44.152 IST
Key label: ncs1k
Type     : RSA General purpose
Size     : 2048
Created  : 10:02:32 IST Tue Jun 27 2017
Data     :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 00B59CFD DF2A2AEF B5B3DB63 AED2F5CB 9C02E519 E379099C F24543E0 4F19310F
 1F7B981E 3520C20E D7934082 D9BF04D7 07E5824F EA5EA1BB DDDFF6CD 9FCBDF75
 F3EF39DA C08B7C69 40AC89D4 58FF3FD0 1ED2AC8C 770C2339 E8508B48 E648A15D
 FE3DE9FA 05E878B2 3094E2F8 8F6280C3 469FF22F 386483FC 5EDE8178 5F7537C6
 7B5C487E 7E6BC636 2EBC55E4 3A6264CE A113BE64 A20F47F0 E1AA603E D5DB078F
 A0B0F36E 4314C435 2283D93F 40B4FEEE 63C33968 DB399B2C 88D97ADE F8DF6ED9
 2CAD24BD FA86CF36 247DE466 E2622D79 B25779D3 FADFDDE2 70474236 2FD58F5A
 67D6CC24 38DE7C8F 33923479 E822E92D C9B141FF E576C59C 50BB3CC5 F693A7D4
 81020301 0001
```

Associated Commands

- crypto key generate rsa

Configure Trust Point

configure

crypto ca trustpoint {*ca-name*}

enrollment url {*ca-url*}

subject-name {*x.500-name*}

rsakeypair{*keypair-label*}

crl optional

exit

commit

Examples

The following is a sample of configuring the trust point.

```
configure
crypto ca trustpoint ncs1k
enrollment url http://209.165.200.226
subject-name CN=ncs1k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
rsakeypair ncs1k
crl optional
exit
commit
```

The following is sample output of **show run crypto ca trustpoint** *ca-name* command.

```
Tue Jun 27 10:39:40.375 IST
crypto ca trustpoint ncs1k
 crl optional
 subject-name CN=ncs1k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
 enrollment url http://209.165.200.226
 rsakeypair ncs1k
!
```

Associated Commands

- crypto ca truspoint
- enrollment url
- rsakeypair

Authenticate Certificate Authority and Request Certificates

This procedure authenticates the certificate authority (CA) with NCS 1002 and requests certificates from the CA. NCS 1002 must authenticate the CA by obtaining the self-signed certificate of the CA. The self-signed certificate contains the public key of the CA. It is required to manually authenticate the public key of the CA by contacting the CA administrator to compare the fingerprint of the CA certificate.

configure

crypto ca authenticate {*ca-name*}

crypto ca enroll {*ca-name*}

exit

commit

Examples

The following is a sample of authenticating the certificate authority and requesting certificates.

```
configure
crypto ca authenticate ncs1k
crypto ca enroll ncs1k
exit
commit
```

The following is sample output of **show crypto ca certificates** command.

```
Tue Jun 27 10:39:47.356 IST

Trustpoint       : ncs1k
================================================
CA certificate
  Serial Number  : 01
  Subject:
        CN=ncs1k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
  Issued By      :
        CN=ncs1k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
  Validity Start : 03:50:38 UTC Wed Mar 22 2017
  Validity End   : 03:50:38 UTC Sat Mar 21 2020
  SHA1 Fingerprint:
         0B2E1F69BB42CE068AAB67F1C2C09C9FAF5F3F66
Router certificate
  Key usage      : General Purpose
  Status         : Available
  Serial Number  : 01:1F
  Subject:
        serialNumber=cf302761,unstructuredAddress=209.165.200.226,unstructuredName=ncs1k,
        C=US,ST=NY,L=Newyork,O=Govt,OU=BU,CN=ncs1k
  Issued By      :
        CN=ncs1k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
  Validity Start : 04:37:11 UTC Tue Jun 27 2017
  Validity End   : 04:37:11 UTC Wed Jun 27 2018
  SHA1 Fingerprint:
         AF640E4E9E0521217BF3F4770465FD832B2AE90B
Associated Trustpoint: ncs1k
```

Associated Commands

- crypto ca authenticate
- crypto ca enroll

Configure EAP Profile

You can configure multiple EAP profiles.

configure

eap profile {*name*}

identity {*user-name*}

method tls pki-trustpoint {*trustpoint-name*}

exit

commit

Examples

The following is a sample of configuring the EAP profile.

```
configure
eap profile ncs1k
identity PRO67
method tls pki-trustpoint ncs1k
exit
commit
```

The following is sample output of **show run eap** command.

```
Tue Jun 27 10:39:51.211 IST
eap profile ncs1k
 method tls
  pki-trustpoint ncs1k
 !
 identity PRO67
!
```

Associated Commands

- eap profile
- pki-trustpoint

Configure 802.1X Profile

You can configure multiple 802.1X profiles. The role of the node running 802.1X profile can be supplicant, authenticator, or both.

configure

dot1x profile {*name*}

pae {**authenticator** | **supplicant** | **both**}

authentication timer reauthenticate {*seconds* | **server**}

supplicant eap profile {*profile-name*}

exit

commit

Examples

The following is a sample of configuring the 802.1X profile.

```
configure
dot1x profile reauth
pae both
authentication timer reauthenticate 3600
supplicant eap profile ncs1k
exit
commit
```

The following is sample output of **show run dot1x** command.

```
Tue Jun 27 10:39:55.178 IST
dot1x profile both_local_reauth
 pae both
 authenticator
  timer reauth-time 3600
 !
 supplicant
  eap profile ncs1k
 !
!
```

Associated Commands

- dot1x profile
- authentication timer reauthenticate
- dot1x supplicant eap profile

Configure EAP and 802.1X Profile on MACsec Controller

You can attach one of the 802.1X profiles on the MACsec controller.

configure

controller MACSecCtrl *Rack/Slot/Instance/Port*

dot1x profile *profile-name*

macsec eap [**policy** *macsec-policy-name*]

exit

commit

Example

The following is a sample of configuring MACsec EAP and 802.1X profile on the MACsec controller.

```
configure
```

```
controller MACSecCtrl 0/0/0/24
dot1x profile reauth
macsec eap
exit
commit
```

Associated Commands

- dot1x profile
- macsec eap

Verify EAP and 802.1X Configuration on MACsec Controller

show run controller mACSecCtrlr *Rack/Slot/Instance/Port*

```
Tue Jun 27 10:39:59.148 IST
controller MACSecCtrlr0/0/0/24
 dot1x profile both_local_reauth
 macsec eap
!
```

show dot1x controller mACSecCtrlr *Rack/Slot/Instance/Port* **detail**

```
Tue Jun 27 10:40:02.648 IST

Dot1x info for MACSecCtrlr0/0/0/24
----------------------------------------------------------------
Interface short name      : MS0/0/0/24
Interface handle          : 0x8000544
Interface MAC             : 2c00.4314.5c6c
Ethertype                 : 888E
PAE                       : Both
Dot1x Port Status         : AUTHORIZED
Dot1x Profile             : both_local_reauth
Supplicant:
   Config Dependency      : Resolved
   Eap profile            : ncs1k
   Client List:
      Authenticator       : 2c02.dc14.636c
      EAP Method          : EAP-TLS
      Supp SM State       : Authenticated
      Supp Bend SM State  : Idle
      Last authen time    : 2017 Jun 27 209.165.200.227
Authenticator:
   Config Dependency      : Resolved
   ReAuth                 : Enabled, 0 day(s), 01:00:00
   Client List:
      Supplicant          : 2c02.dc14.636c
      Auth SM State       : Authenticated
      Auth Bend SM State  : Idle
      Last authen time    : 2017 Jun 27 209.165.200.225
      Time to next reauth : 0 day(s), 00:46:09
MKA Interface:
   Dot1x Tie Break Role   : Auth
   EAP Based Macsec       : Enabled
   MKA Start time         : 2017 Jun 27 209.165.200.227
   MKA Stop time          : NA
   MKA Response time      : 2017 Jun 27 209.165.200.226
```

show macsec mka session controller mACSecCtrlr *Rack/Slot/Instance/Port*

```
Tue Jun 27 10:40:07.492 IST


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

Local Tx-SCI            : 2c00.4314.5c6c/0018
Local Tx-SSCI           : 2
Interface MAC Address   : 2c00.4314.5c6c
MKA Port Identifier     : 24
Interface Name          : MS0/0/0/24
CAK Name (CKN)          : A3B6509EC0EBCAE8D610139669952E0A
CA Authentication Mode  : EAP
Keychain                : NA (EAP mode)
Member Identifier (MI)  : C12DD7CF165438D8B1732211
Message Number (MN)     : 422
Authenticator           : YES
Key Server              : YES
MKA Cipher Suite        : AES-128-CMAC

Latest SAK Status       : Rx & Tx
Latest SAK AN           : 0
Latest SAK KI (KN)      : C12DD7CF165438D8B173221100000001 (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)
Time to SAK Rekey       : NA

MKA Policy Name         : *DEFAULT POLICY*
Key Server Priority     : 16
Delay Protection        : FALSE
Replay Window Size      : 64
Include ICV Indicator   : FALSE
Confidentiality Offset  : 0
Algorithm Agility       : 80C201
SAK Cipher Suite        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired          : YES

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

Live Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
  578656B568B072819160DCD4         420  2c02.dc14.636c/0018     1          16

Potential Peer List:
  MI                        MN          Rx-SCI (Peer)        SSCI KS-Priority
  ------------------------------------------------------------------------------
```

show macsec mka session

```
Tue Jun 27 10:40:59.320 IST
NODE: node0_RP0_CPU0
==============================================================================
      Interface         Local-TxSCI       # Peers   Status  Key-Server
```

```
=============================================================================
     MS0/0/0/24      2c00.4314.5c6c/0018      1     Secured    YES
```