Technical Guide

MACsec

Feature Overview and Configuration Guide

Introduction

This guide describes MACsec (Media Access Control Security) and how to configure it.

MACsec provides line-rate encryption and protection of traffic passing over a Layer 2 network or link. It protects all frames passing over the link, including Layer 2 protocols such as ARP. MACsec can provide the following services:

- Connectionless data integrity—ensures the frame has not been modified en route.
- Data origin authenticity—ensures the frame was sent by one of the MACsec peers.
- Confidentiality—encrypts the frame's EtherType and payload to ensure they cannot be read en route.
- Replay protection—ensures the same frame is not received more than once.

Note that MACsec operates within a single Layer 2 network or segment, so it cannot provide end-to-end protection of routed IP traffic, such as traffic passing over the open Internet.

This Guide provides:

- an "Overview of MAC Security (MACsec)" on page 4
- a description of "How MACsec Works" on page 5
- "Packet Overheads and Throughput Limits" on page 11
- information about "Configuring MACsec" on page 13, including a step by step procedure and examples.

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support MACsec (MAC Security), running version **5.4.9-2** or later.

Table 1: Products that support MACsec

SUPPORTED HARDWARE			SUPPORTED CIPHER SUITES	
SWITCH	XEM	PORTS	GCM-AES-128	GCM-AES-256
IE360		2x uplink SFP+ ports at 1 and 10G speeds	Yes (from first release)	Yes (from first release)
IE560		All ports	Yes (from first release)	Yes (from first release)
x550-18XSQ		1-16 (SFP+ ports)	Yes (from 5.5.1-2.1)	—
x930		Front panel 1G ports	Yes (from 5.4.9-2.1)	—
x950 and SBx908 GEN2	XEM2-12XS	All ports	Yes (from 5.4.9-2.1)	—
	XEM2-12XS v2	All ports	Yes (from 5.5.1-1.1)	Yes (from 5.5.1-2.1)
	XEM2-8XSTm	5-8 (SFP+ ports)	Yes (from 5.5.1-1.1)	Yes (from 5.5.1-2.1)
SBx908 GEN3	XEM3-2DQ	All ports	Yes (from 5.5.6-0.1)	Yes (from 5.5.6-0.1)
	XEM3-8CQ	All ports	Yes (from 5.5.6-0.1)	Yes (from 5.5.6-0.1)
	XEM3-12YS	All ports	Yes (from 5.5.6-0.1)	Yes (from 5.5.6-0.1)

From version 5.5.1-2.1, all MACsec-capable products support 256-bit pre-shared Secure Connectivity Association Keys (CAKs).

For more information, see the product's Command Reference.

This document is available from our website at alliedtelesis.com.

Overview of MAC Security (MACsec)

AlliedWare Plus supports MACsec with the MACsec Key Agreement protocol (MKA) and pre-shared keys. The process works like this:

- The network administrator configures a pre-shared key on each device. This is known as the Secure Connectivity Association Key (CAK).
- Each device automatically discovers its peers through the MACsec Key Agreement protocol (MKA). They use the CAK for mutual authentication, that is, to prove that each device is a legitimate peer and not an imposter.
- MKA randomly generates and distributes new encryption keys, known as Secure Association Keys (SAKs), to all devices.
- MACsec uses the SAKs to encrypt and verify frames passing over the protected link.

When MKA/MACsec is configured on a particular switchport, it immediately blocks the port. No Ethernet frames can ingress or egress through the port except for the MACsec Key Agreement protocol. It only unblocks the port when MKA has discovered peers and has distributed SAKs. Then MACsec protects the data passing over the link.

Standards

The MACsec and MKA protocols are described in the following IEEE standards:

- IEEE 802.1X-2010 describes MACsec Key Agreement (MKA), a protocol to discover peers and distribute encryption keys.
- IEEE 802.1AE-2006 describes the method of protecting traffic by encapsulating it inside MACsec frames. This uses the encryption keys distributed through MKA.

When to use MACsec or IPsec

Use MACsec to protect all frames passing over a link in a single Layer 2 network, including Layer 2 protocols such as ARP. MACsec operates at Layer 2 only.

If IP packets are being routed between different L2 networks, then MACsec cannot provide end-to-end protection; frames must be decrypted and re-encrypted when they are routed. IPsec can be used to protect Layer 3 IPv4 and IPv6 traffic passing through the Internet. For information about IP Security, see the Internet Protocol Security (IPsec) Feature Overview and Configuration Guide.

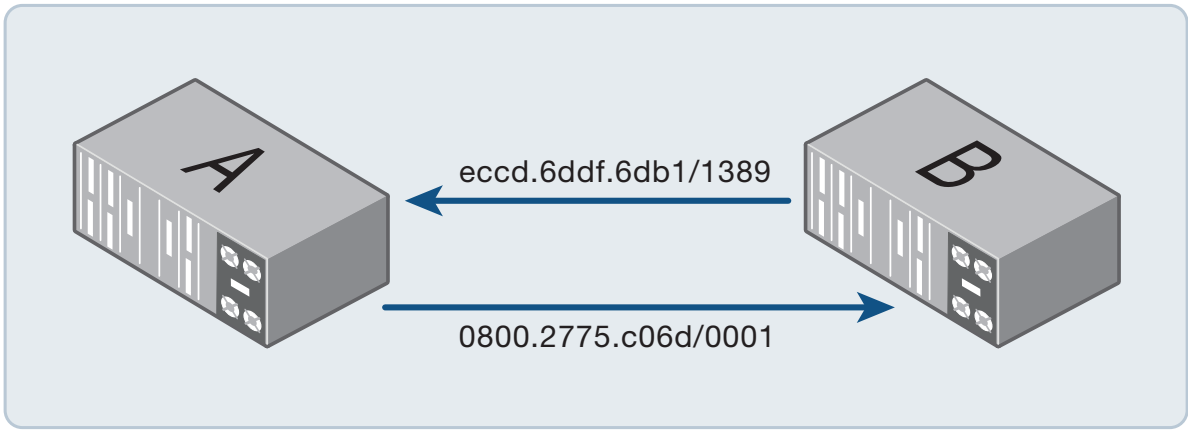
How MACsec Works

MACsec concepts

Each MACsec participant has a **Secure Channel** (SC) that it uses to send traffic to other participants. Each channel is one-directional; one participant uses it to send traffic and other participants receive that traffic.

Each channel has an 8-byte **Secure Channel Identifier** (SCI). The first 6 bytes match the MAC address of the device transmitting through that channel. The remaining 2 bytes are a 'Port Identifier' used to distinguish between multiple channels from the same device.

Figure 1: Secure Channels between two devices, labeled with their Secure Channel Identifiers



Each channel may contain 0-2 **Secure Associations** (**SAs**) at any point in time. These contain the following pieces of information, which are required to protect and verify frames being sent and received through the channel:

- An encryption key, known as the **Secure Association Key** (SAK).
- Counters related to packet numbers.

Most of the time, once MACsec is working, a channel will only have one SA. However, an SA needs to be replaced with a new one from time to time and the channel will briefly have two SAs while swapping from one to the other.

Each MACsec-protected frame contains a 'packet number'. Packet numbers are used by the encryption and verification process and they are also used for replay protection. At the transmission end of the channel, the SA contains a counter to record which packet number will be used next. At the receiving end, the SA keeps a record of which packet number it expects to see next. The first frame to be protected with a particular SA has packet number one and it increases by one for each subsequent frame.

An SA is identified through a combination of the channel's SCI and an automatically-assigned **Association Number** (AN). There are only four possible values for an Association Number (0-3), so ANs are reused regularly.

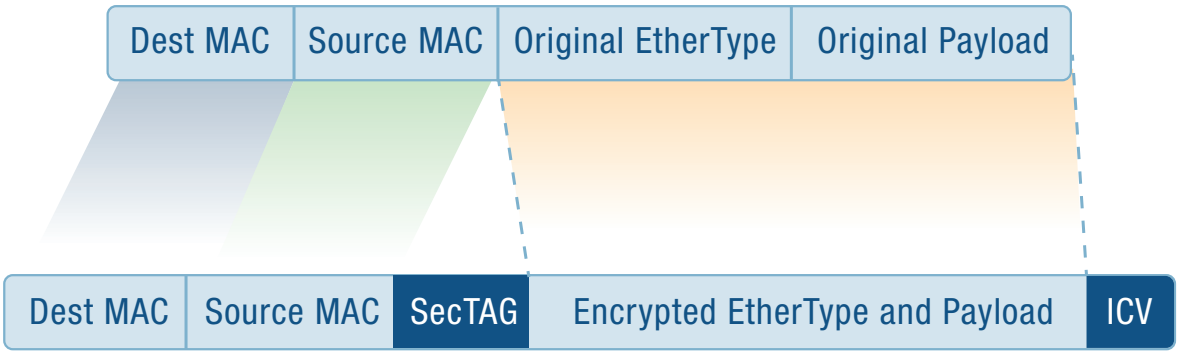
MACsec encapsulation

MACsec modifies each frame that passes over the link in the following ways:

- It inserts a MACsec tag (**SecTAG**) just before the EtherType.
- It adds an Integrity Check Value (**ICV**) at the end of the frame.
- It **encrypts** the original EtherType and frame payload (the bytes after the SecTAG and before the ICV).

These changes add 32 bytes to the frame-size. (The SecTAG is 16 bytes and the ICV is 16 bytes.)

Figure 2: Frame with MACsec encapsulation



The MACsec peer receiving the encapsulated frame will check the information in the SecTAG and the ICV to determine whether to drop or accept and decrypt the frame.

Only legitimate MACsec participants know how to calculate the ICV because the calculation uses the Secure Association Key. The ICV is also derived from the frame contents, including the MAC addresses and the SecTAG. If a frame is modified en route, or a third party tries to inject frames into the network, then the ICV will be incorrect and the frame will be discarded. Thus, the MAC addresses, the EtherType and the payload are all protected from modification.

Note that IEEE 802.1AE allows for a VLAN tag to be inserted before the SecTAG. In that case the VLAN tag would not be included in the ICV calculation and would not be protected from modification en route. AlliedWare Plus does not support this option—VLAN tags are always placed after the SecTAG, so they are encrypted and part of the ICV calculation.

The SecTAG starts with the MACsec EtherType (0x88E5) so that MACsec-protected frames can be distinguished from unprotected frames. The SecTAG also contains other important contextual information including:

- The SCI of the channel the frame is being sent through.
- The Association Number of the SA used to protect the frame.
- The packet number.
- An indication of whether encryption was applied. (In AlliedWare Plus encryption is always applied).

Replay protection

When replay protection is enabled, MACsec drops frames that are too far out of the expected order. This protects against potential replay attacks that could copy a legitimate MACsec-protected frame and repeatedly transmit it into the network. Without protection, such an attack could potentially cause difficulties by flooding the receiving network or by getting protocols to behave in unintended ways.

MACsec transmits each frame in an SA with a packet number starting from one and increasing by one for each frame. The peer receiving a frame compares its number with its 'lowest acceptable PN' and drops any frames with lower numbers.

The replay window determines whether the receiving MACsec peer will only accept frames in the same order as they were transmitted, or will allow some frame reordering. It expects the next packet number to be one greater than the highest packet number it has received so far. The lowest acceptable packet number is this expected next packet number less the configured replay window size:

```
<lowest-acceptable-PN> = <next-PN> - <replay-window>
```

- If you set the replay window to 0, it will allow no frame reordering. MACsec will only accept packet numbers higher than it has seen before. This provides the greatest protection against replay attacks, but if frames normally arrive out of order, this may drop legitimate frames.
- If you set a replay window > 0, this will allow limited reordering within the replay window and drop frames from earlier than the replay window. This allows the possibility of receiving the same frames more than once, but only for the most recent frames.

AlliedWare Plus does not support delay protection, which would otherwise also use these packet number counters.

MKA peer discovery and authentication

The MACsec Key Agreement protocol (MKA) automatically discovers peer devices and verifies that they are legitimate.

All MKA participants need to have matching configuration for the following:

- the Secure Connectivity Association Key (CAK)—an encryption key
- the CAK name—a name that's included in each MKA message to identify which CAK is being used.

Each device sends MKA messages at regular intervals to the PAE group MAC address (0180.c200.0003). MKA uses the EAPOL EtherType (0x888e). This is the same EtherType as 802.1X EAP port authentication. (See AAA and Port Authentication Feature Overview and Configuration Guide.)

The MKA messages include the sender's own member ID and the IDs of other potential peers that it has received messages from. Each device randomly generates its own member ID when the protocol is started. That ID is different each time the protocol is restarted.

When a device receives an MKA message, it verifies that the sender has the correct CAK and CAK name. If they don't match the receiver's configuration then it ignores the message. The device considers a peer to be live when it sees that the peer has also seen it.

When a device does not receive a valid message from a peer for 6 seconds, it assumes that the peer is gone. It removes the peer's channel from hardware and it will no longer be able to receive MACsec-protected traffic from that peer.

MKA key server

One device is elected to act as the 'key server'. The key server:

- decides the MACsec protection settings to use (including which cipher suite to use)
- generates and distributes the Secure Association Keys (SAKs)
- initiates and coordinates changeovers from one Secure Association to the next.

When an AlliedWare Plus device is acting as key server, it will always pick the cipher suite specified by the **macsec-cipher-suite** command. When the other device runs AlliedWare Plus, it will use whatever settings the key server has chosen, so long as those settings are supported. However, if the key server chooses settings that are not supported then MACsec will not unblock the port. Beware that different products and software versions support a different range of cipher suites as shown in Table 1 on page 3.

With AlliedWare Plus, MKA always uses pre-shared keys (CAKs), so the key server is elected by comparing priority values configured on each device. Key-server priority values range from 0 (highest priority) to 255 (lowest priority). If two devices have the same priority, then they compare the SCIs of their transmit channels to elect the key server.

Supported features and limitations

AlliedWare Plus supports MACsec in combination with the MACsec Key Agreement protocol (MKA) and pre-shared Secure Connectivity Association Keys (CAKs).

- MACsec/MKA cannot be used in combination with IEEE 802.1X EAP authentication.
- MACsec/MKA can only be configured on switchports that are not members of static or dynamic link aggregations.
- Each MACsec-protected switchport can only have one MKA peer. AlliedWare Plus does not support two or more peers on a single port.
- MKA messages are always addressed to the PAE group MAC address (0180.c200.0003).
- The CAK can be either 128 bits or 256 bits on all MACsec-capable products. (This key is used to secure MKA messages and authenticate MKA peers. It is not used for encryption of general link traffic.)
- AlliedWare Plus lets you use different key sizes for the CAK and the MACsec cipher suite. That means you can use a 256-bit CAK when the MACsec cipher suite is GCM-AES-128.
- Refer to Table 1 on page 3 to see which MACsec cipher suites are supported on each product. All MACsec-capable products support GCM-AES-128 and some also support GCM-AES-256. AlliedWare Plus does not support the cipher suites with extended packet numbering (GCM-AES-XPN-128 and GCM-AES-XPN-256). The MACsec cipher suite is used for applying MACsec protection to general link traffic.
- AlliedWare Plus only supports integrity protection with confidentiality for MACsec encapsulation. That is, frames are protected from modification and the payload is encrypted. Integrity protection without confidentiality is not supported. Confidentiality offsets are not supported.
- MKA cannot be used without MACsec protection of traffic.
- If the frames to be protected by MACsec are VLAN-tagged, then the VLAN tags are included inside the MACsec encapsulation. That is, the SecTAG is inserted before the VLAN tag. This means the VLAN tag is encrypted and protected from modification.
- Delay protection is not supported.
- On x550 and x930, extra configuration is required to ensure important control packets are not dropped when the link is oversubscribed. That extra configuration may result in less throughput than would be ideal. See "Packet Overheads and Throughput Limits" on page 11.

Interactions with other features

Maximum Receive Unit (MRU)

The extra 32 bytes that are added to a frame when it is protected by MACsec are not counted towards the MRU. You don't need to change the MRU setting on the MACsec interfaces.

VCStacking

MACsec supports stack failover, but internal state machines are not synchronized. During a master failover, the new stack master performs the following steps:

1. It reinitializes the hardware drivers of all MACsec ports so that all non-EAPOL traffic is blocked.
2. It waits 8 seconds for the remote MKA peer to time out its knowledge of the old master as an MKA participant.
3. It restarts MKA on all MACsec ports. Once MKA is restarted, the traffic following is again protected by MACsec.

Port mirroring

For egress traffic, port mirroring is done before any MACsec processing. For ingress traffic, port mirroring is done after any MACsec processing. That means:

- Mirrored frames do not have MACsec encapsulation.
- Egress frames will be mirrored even when MACsec is blocking the port.
- Ingress frames will not be mirrored if they are discarded by MACsec.

Access Control Lists (ACLs)

Hardware ACLs are applied to ingress traffic after any MACsec processing. That means:

- The ACLs do not see any MACsec encapsulation.
- If a frame is blocked by MACsec then ACL actions such as ‘copy to CPU’ will not take place.

Packet Overheads and Throughput Limits

MACsec adds overhead to traffic that affects throughput. On the x550 and x930, you need to limit traffic to make sure that important control packets are not dropped. This section describes:

- "Packet overheads" on page 11—packet sizes with MACsec encapsulation
- "Throughput" on page 11—the effects of MACsec on throughput
- "Congestion on x950 and SBx908 GEN2" on page 12—how these devices deal with congestion
- "Congestion on x930 and x550-18XSQ" on page 12—why you need to limit traffic on MACsec-enabled ports to prevent random packet loss.

Packet overheads

MACsec encapsulation adds 32 bytes to each encrypted frame that it sends over the wire at the physical layer (Layer 1) ("MACsec encapsulation" on page 6). This is in addition to the standard 20 bytes the device already adds to frames, increasing the overhead on the physical wire (Layer 1) from 20 bytes to 52 bytes. This reduces the effective traffic throughput. The smaller the frames, the larger proportion of bandwidth is needed for overheads. For the smallest frames (64 bytes), this means:

Table 2: Packet overhead on 64-byte frames with and without MACsec encapsulation (in bytes)

MACsec	Layer 2 Ethernet frame-size	MACsec overhead	Layer 1 overhead	Total packet size
No	64	-	20	84
Yes	64	32	20	116

Throughput

On a standard 1Gbps link without MACsec encryption, this means that for 64-byte Layer 2 frames, the link can transmit 1 488 095 frames per second. The same link with MACsec encryption enabled can only transmit 1 077 586 frames per second.

Table 3: Bandwidth composition and throughput for 64-byte frames with and without MACsec (in Mbps)

MACsec	64-byte Layer 2 Ethernet frames per second	Layer 2 Ethernet frames (Mbps)	MACsec overhead (Mbps)	Layer 1 overhead (Mbps)	Total bandwidth (Mbps)
No	1 488 095	761.9	-	238.1	1000
Yes	1 077 586	551.7	275.9	172.4	1000

For larger-sized frames, the overhead transmitted per second is lower and the effective throughput per second is higher, because the fixed frame-overheads are applied to fewer frames:

Table 4: Cumulative throughput on a 1Gbps link with MACsec enabled at selected frame-sizes

MACsec	Frame-size (bytes)	L2 Ethernet frame throughput (Mbps)	L1 Ethernet packet throughput (Mbps) (includes 20 bytes per frame overhead)	MACsec encrypted L1 Ethernet packet throughput (Mbps) (includes 52 bytes per frame overhead)
Yes	64	551.7 Mbps	724.1 Mbps	1000 Mbps
Yes	128	711.1 Mbps	822.2 Mbps	1000 Mbps
Yes	512	907.8 Mbps	943.3 Mbps	1000 Mbps
Yes	1518	966.9 Mbps	979.6 Mbps	1000 Mbps

In real networks, the size of Ethernet frames is mixed, generally with most frames sized 64-128 bytes and 1024-1518 and fewer frames in the other size ranges. The exact mix of frame-sizes in your network will depend on the applications running and will change depending on the nature of the traffic crossing the network at any moment in time. On an individual link, the mix of frame-sizes may also differ substantially between transmitted frames and received frames.

Because of the overhead, with MACsec enabled, the usable bandwidth on the link is reduced to the levels shown in the middle column of Table 4 (L2 Ethernet frame throughput). If the device attempts to send more traffic than this over the link, frames will be dropped. The manner in which frames are selected to be dropped depends on the platform, as follows.

Congestion on x950 and SBx908 GEN2

On the x950 and SBx908 GEN2 switches, when traffic reaches the bandwidth limit of a MACsec-enabled port (Table 4, column 3), the switch will apply its normal egress priority queuing rules. It prioritizes important network control packets, so the control frames for such protocols as STP, EPSR, OSPF and MKA (the MACsec control protocol itself) are not discarded.

Congestion on x930 and x550-18XSQ

On the x930 and x550-18XSQ switches, when traffic reaches the bandwidth limits of a MACsec-enabled port on an x930 switch (Table 4, column 3), the device discards some packets at random with no priority queuing. This means that on these switches with default configuration, MACsec-enabled ports may randomly drop important control packets for network protocols such as STP, EPSR, OSPF and MKA (protocol controlling MACsec). This can disrupt these protocols and potentially lead to network disruptions, such as loops formed by STP or EPSR, losing OSPF neighbor relationships or MACsec restarting and blocking all traffic on the port for 5-6 seconds while it renegotiates its MKA sessions.

For information about limiting traffic on the x930 and x550-18XSQ switches to prevent network disruptions arising from congestion, see "Avoid control packet loss on x930 and x550" on page 18.

Configuring MACsec

Configuration procedure

To configure MACsec, you follow these steps:

- Step 1: **Enable MACsec hardware support**
- Step 2: **Create an MKA policy**
- Step 3: **Add a pre-shared key (CAK) to the interface**
- Step 4: **Add the MKA policy and enable MACsec protection on the port**
- Step 5: **Control egress traffic rate**
- Step 6: **Verify MACsec configuration**

Step 1: Enable MACsec hardware support

This step is only required for some device types as shown in Table 5. If your device is marked 'Yes' then enter the commands listed in this step. Otherwise, go to step 2.

Table 5: Devices that require **platform macsec enable** command

Switch	XEM	Requires 'platform macsec enable' command?
x550-18XSQ		Yes
x930		Yes
x950 and SBx908 GEN2	XEM2-12XS	Yes
	XEM2-12XS v2	—
	XEM2-8XSTm	—

Enable hardware support for MACsec on the device.

```
awplus#configure terminal
awplus(config)#platform macsec enable
```

You will need to save the configuration and restart the device before this will take effect and before you can enter any more MACsec configuration.

```
awplus#write
awplus#reboot
reboot system? (y/n): y
```

Step 2: Create an MKA policy

Either create a new MKA policy, or use the default policy if it suits your network. To create a policy:

```
awplus(config)#mka policy <policy-name>
```

Note that to create the policy, the device must be in Global Configuration mode. (The same command syntax is used in Interface Configuration mode to add the policy to a port.)

Replay protection

By default, replay protection is enabled with a replay window of 0. We recommend using the default setting unless you expect legitimate frame reordering on the link.

```
awplus(config-mka-policy)#macsec replay-protection window-size <0-
4294967295>
```

Key server priority

By default, the key-server priority is set to 128. To make this device more or less likely to become the key server, you can change its priority—a lower number has a higher priority.

```
awplus(config-mka-policy)#key-server priority <0-255>
```

MACsec cipher suite

The device that is elected as key server gets to choose which cipher suite MACsec uses to protect traffic.

By default, AlliedWare Plus chooses GCM-AES-128 when elected as key server. If your device supports GCM-AES-256 then you can configure it to choose that cipher suite instead. To see which cipher suites your device supports, refer to Table 1 on page 3.

```
awplus(config-mka-policy)#macsec-cipher-suite gcm-aes-256
awplus(config-mka-policy)#exit
```

Step 3: Add a pre-shared key (CAK) to the interface

This step configures the following (notes on choosing these follow):

- The Secure Connectivity Association Key (CAK).
  An encryption key used to secure MKA messages and authenticate MKA peers.
- The CAK Name (CKN).
  A label that's included in each MKA message to identify which CAK is being used.

To configure the pre-shared key for the interface, use the commands:

```
awplus(config)# interface <port>
awplus(config-if)# mka pre-shared-key ckn <cak-name> cak <cak>
```

Notes—choosing the CAK

Choose the size of the CAK from these options:

- 128 bits (16 bytes), entered as 32 hexadecimal characters.
- 256 bits (32 bytes), entered as 64 hexadecimal characters.

We recommend you always use a 256-bit CAK, regardless of which MACsec cipher suite you choose (with the **macsec-cipher-suite** command).

The CAK should be a value from a cryptographically secure random number generator. We recommend you use the **crypto random bytes** command. To create a 256-bit CAK, use the command:

```
awplus# crypto random bytes 32
```

The **crypto random bytes** command prints a random number to the console; it does not store or configure the number anywhere. To use the number as a key, you need to copy it and enter it in the **mka pre-shared-key** command.

Notes—choosing the CAK name

- It can be 2 to 64 hexadecimal digits long (an even number of digits, representing 1 to 32 bytes).
- It must not be derived from any part of the CAK because that would seriously undermine protection. (The CKN is clearly visible in every MKA message; it is not encrypted.)
- Each different CAK used in your network should have a unique name.

Step 4: Add the MKA policy and enable MACsec protection on the port

```
awplus(config-if)#mka policy <policy-name>
```

To use the default policy, enter 'default' as the policy name. Otherwise, enter the name of the policy you created in Step 2.

Note that to add the policy to the interface, the device must be in Interface Configuration mode. (The same command syntax is used in Global Configuration mode to create and enter configuration mode for the policy.)

Step 5: Control egress traffic rate

This step is only required for some device types as shown in Table 6. For other devices, skip this step.

Table 6: Device types that require configuration to control egress traffic rate

Device	Requires configuring egress traffic rate?
x550-18XSQ	Yes
x930	Yes
x950	—
SBx908 GEN2	—

On these devices, you must limit traffic transmitted to prevent the loss of control frames, including MKA frames, that could otherwise interrupt traffic. Choose one of the following options.

- Option A: egress-rate-limit command (recommended)

  The commands below will work for all frame sizes when the switchport is running at its top speed. Consider your network traffic profile—you may be able to tune this for greater link utilization. For more information, see "Avoid control packet loss on x930 and x550" on page 18.

On the **x550**:

```
awplus(config)# interface <port>
awplus(config-if)# egress-rate-limit 5517m
```

On the **x930**:

```
awplus(config)# interface <port>
awplus(config-if)# egress-rate-limit 551700k
```

- Option B: Flow control (x930 only)

  If flow control is feasible across the whole network, you may be able to use it to limit bandwidth on an x930. Flow control is supported on the x930 but it is not supported on the x550. For more information on requirements for using flow control, see "Avoid control packet loss on x930 and x550" on page 18.

  To enable flow control on all ports of an x930, use the commands:

```
awplus(config)# interface <port-list>
awplus(config-if)# flowcontrol receive on
awplus(config-if)# flowcontrol send on
```

Step 6: **Verify MACsec configuration**

Check the MKA policy settings.

```
awplus# show mka policy [<policy-name>]
```

The following example output shows these policies:

- A user-configured policy (‘P256’). The cipher suite is set to GCM-AES-256.
- The inbuilt default policy (‘default’).

Output 1: Example output from **show mka policy**

```
awplus#show mka policy
MKA policy: P256
  Key server priority: 128
  MACsec cipher suite: GCM-AES-256
  Replay protection: Enabled
  Replay window size: 0

MKA policy: default
  Key server priority: 128
  MACsec cipher suite: GCM-AES-128
  Replay protection: Enabled
  Replay window size: 0
```

Configuration example: MACsec with default MKA policy

This section shows how to configure MACsec with the default MKA profile on different kinds of hardware. Where an example uses the **egress-rate-limit** command, it assumes the following:

- The port is running at its top speed.
- It needs to work with all frame sizes, including the worst case scenario of all frames being 64 bytes.

In the examples below, ***<64-hexadecimal-characters>*** is a placeholder for the pre-shared CAK. We recommend you use the following command to generate a random 256-bit key to use as the CAK. Both ends of the link need the same pre-shared CKN and CAK.

```
awplus# crypto random bytes 32
```

The output of this command is 64 hexadecimal characters. Copy and paste this into the **mka pre-shared-key** command below.

On x550

```
platform macsec enable

interface port1.0.1
  mka pre-shared-key ckn 01 cak <64-hexadecimal-characters>
  mka policy default
  egress-rate-limit 5517m
```

On x930

```
platform macsec enable

interface port1.0.1
  mka pre-shared-key ckn 01 cak <64-hexadecimal-characters>
  mka policy default
  egress-rate-limit 551700k
```

On x950 or SBx908 GEN2 with XEM2-12XS

```
platform macsec enable

interface port1.1.1
  mka pre-shared-key ckn 01 cak <64-hexadecimal-characters>
  mka policy default
```

On x950 or SBx908 GEN2 with XEM2-12XS v2 or XEM2-8XSTm

```
interface port1.1.5
  mka pre-shared-key ckn 01 cak <64-hexadecimal-characters>
  mka policy default
```

Configuration example: MACsec using GCM-AES-256

On x950 or SBx908 GEN2 with XEM2-12XS v2 or XEM2-8XSTm

```
mka policy P256
  macsec-cipher-suite gcm-aes-256

interface port1.1.1
  mka pre-shared-key ckn 01 cak <64-hexadecimal-characters>
  mka policy P256
```

Avoid control packet loss on x930 and x550

On an x930 or x550, to avoid dropping control packets (including MKA frames that control MACsec), with associated disruptions to the network, you must limit the traffic the port attempts to transmit.

This section provides information for choosing a suitable configuration:

- You can use "Egress-rate-limiting" on page 18 on MACsec-enabled ports on an x930 or x550 to limit bandwidth and prevent control packet loss by MACsec. This may limit bandwidth more than necessary.
- "Calculating an egress-rate-limit" on page 19 may allow you to tune the limit for bandwidth utilisation.
- On the x930, "Flow control (x930)" on page 20 on all ports in the network can provide greater link utilization. This is a suitable solution only if it is feasible to use flow control over the whole network. (Flow control is not supported on the x550.)

For the commands used to configure the solution, see the "Configuration procedure" on page 13. For more information about why this is necessary, see "Packet Overheads and Throughput Limits" on page 11.

Egress-rate-limiting

You can use egress-rate-limiting on the MACsec port to limit traffic to a safe level on an x550 or an x930. When egress-rate-limiting is applied at a suitable level, the switch discards frames while applying priority queuing rules. This means important control packets are not dropped. However, you must choose a fixed value to limit the bandwidth to—it is not dynamically updated with changing network traffic. This may at times artificially reduce the bandwidth below the level that could be safely transmitted out of the port.

To use this method, limit the egress rate to a value corresponding to the ‘L2 Ethernet frame throughput’ (Table 4 on page 12, column 3). You can choose a limit that is low enough to ensure packets are never randomly discarded regardless of frame-size, even if all or most frames are 64 bytes. Such a limit is 551700k on the x930, and 5517m on the x550.

If the average frame-size on the MACsec link is higher than 64 bytes, you may be able to increase the egress-rate-limit to allow greater link utilization. If you set the limit too high for your network traffic, there is a risk of congestion that can lead to randomly dropping important control packets. If a burst of smaller frames are received, this increases the risk.

Calculating an egress-rate-limit

The following equation shows a way to calculate an egress-rate-limit to set on a link with a 5% safety buffer. This is likely to avoid dropping important packets.

$$\text{egress\_rate\_limit\_in\_kbps} = \frac{\text{frame\_size\_in\_bytes}}{\text{frame\_size\_in\_bytes} + 52} \times \text{link\_speed\_in\_kbps} \times 0.95$$

For example, assuming:

- The port is a 1Gbps switchport, as on an x930.
- The average frame-size is 800 bytes.

You can calculate a suitable egress rate limit like this:

$$\text{egress rate limit in kbps} = \frac{800}{800 + 52} \times 1\,000\,000 \times 0.95 = 892018 \text{ kbps}$$

Then configure the egress limit for the port by using this command:

```
awplus(config-if)# egress-rate-limit 892000k
```

If you set the limit too high for your network, there is a risk of congestion that can lead to randomly dropping important control packets.

You can use information from this command to help estimate frame sizes:

```
awplus#show platform port [<port-list>] counters
```

Output 2: Example output from **show platform port counters**

```
awplus#show platform port port1.0.1 counters

Port port1.0.1 Ethernet MAC counters:
 Combined receive/transmit packets by size (octets) counters:
  64                          50272 1024 - MaxPktSz                 0
  65 - 127                   100451 1519 - 2047                     0
  128 - 255                       0 2048 - 4095                     0
  256 - 511                       0 4096 - 9216                     0
  512 - 1023                      0
```

Note that the output from this command:

- displays octet totals for various frame size ranges, not a direct count of frames within each frame size range. For example, 1500 for the 1024-MaxPktSz bucket would represent 1 single frame, whereas 1500 for the 256-511 bucket would represent between 2 and 5 frames.
- combines statistics for traffic received and transmitted. If there are more small frames in one direction than the other, you will not be able to distinguish the transmitted traffic from that received.
- will only be able to give you an estimate of the average frame size due to the limitations noted in the above two bullet points.

Flow control (x930)

You may be able to use flow control instead of or together with egress-rate-limiting if flow control is feasible throughout the network. This allows the switch to instruct its neighbors to pause their data when the traffic load is too high, preventing the x930 from becoming oversubscribed and avoiding it randomly dropping important frames without priority queuing applied. For this method to be effective:

- Flow control must be enabled on all ports on the x930
- flow control must be supported and enabled across the whole network
- the distances to peers must be short enough not to introduce significant delayed responses to flow control frames.

If other devices in the network are not all capable of flow control, then the MACsec link on the x930 may still be oversubscribed, resulting in important dropped packets. Because traffic control only reduces data flow when the load is too high, it can use the full available bandwidth the rest of the time.

Monitoring and troubleshooting MACsec

You can display information and statistics about MACsec for particular interfaces or all interfaces. Use the command:

```
awplus# show macsec [interface <interface-range>]
```

This section provides information for monitoring and troubleshooting MACsec operations, including:

Using output from the ‘show macsec’ command to assess operation

This section goes through a series of things to check when MKA or MACsec are not operating as you expect. For more information about **show macsec** output, see the product’s Command Reference.

In this section:

Is MKA active?

Output 3: Extract from **show macsec** output—MKA is active

```
MKA
  ...
  Active:           True
```

When MKA is active, (‘Active: True’), it means MKA is sending messages and listening for messages from peers. It doesn't indicate whether or not any peers have been discovered.

Output 4: Extract from **show macsec** output—MKA is inactive due to licensing

```
MKA
  ...
  Active:           False (unlicensed)
```

From AlliedWare Plus version 5.5.1-2, when MKA is inactive (‘Active: False’), it may display one of these reasons:

- ‘unlicensed’— Either the device does not have a MACsec license installed or the license is not currently valid according to its start and end dates. Check the **show license external** command to see which licenses are installed.

- ‘incomplete configuration’—MKA is not fully configured. Check that both the commands **mka policy (interface)** and **mka pre-shared-key** are configured on the port.
- ‘interface down’—The switchport is currently down.

It will only show one reason at a time, even if there is more than one reason for MKA being inactive.

Have the devices discovered each other and elected a key server?

Output 5: Extract from **show macsec** output—MKA key server has been elected

```
MKA
  ...
  Key Server SCI:          eccd.6ddf.6db1/1389
```

When things are working, it will display the MKA key server's SCI.

Output 6: Extract from **show macsec** output— MKA key server has not been elected

```
MKA
  ...
  Key Server SCI:          -
```

If a key server has not been elected (‘Key Server SCI: -’) then, most likely, the devices have not discovered each other. Check the following possible reasons:

- Check that MKA messages are being received from the peer. ‘General MACsec Info -> Input EAPOL Pkts’ shows the number of ingress EAPOL frames (including MKA) that have bypassed MACsec verification. See the **show auth statistics interface** command for a counter that includes only MKA messages.

  If there's an intermediary device, it may be intercepting MKA messages. If the peer device is from another vendor then check it is sending MKA messages to the PAE group MAC address (0180.c200.0003).

- Check that both devices have the same CKN and the same CAK. If they don't, both devices will ignore each other.

  AlliedWare Plus always uses the exact CKN and CAK that you have entered into the **mka pre-shared-key** command; it will not pad or truncate your input. If the peer device is from another vendor then check their reference manual to see whether they interpret your input differently.

Has the key server chosen settings that both devices support?

For MACsec to work, the MKA key server needs to choose protection settings that are supported by both devices. The following output is from AlliedWare Plus version 5.5.1-2.

Output 7: Extract from **show macsec** output—MKA key server has chosen acceptable protection settings

```
MKA:
  ...
  MACsec Settings Received:      True
  MACsec Settings Accepted:      True
  MACsec Cipher Suite:           GCM-AES-128
  MACsec Protection Mode:        Integrity and confidentiality
```

If the chosen settings are supported on this switchport then ‘MACsec Settings Accepted’ will display ‘True’.

Output 8: Extract from **show macsec** output—MKA key server has chosen a cipher suite that's not supported on this device

```
MKA:
  ...
  MACsec Settings Received:      True
  MACsec Settings Accepted:      False
  MACsec Cipher Suite:           GCM-AES-256
  MACsec Protection Mode:        Integrity and confidentiality
```

If “MACsec Settings Accepted” displays “False” then at least one of the settings (cipher suite or protection mode) are not supported.

The protection settings, chosen by the key server, are displayed in “MACsec Cipher Suite” and “MACsec Protection Mode”.

Use the **macsec-cipher-suite** command to control which cipher suite is chosen when acting as key server. To see which cipher suites your device supports, refer to Table 1 on page 3.

Are ingress frames being accepted or dropped?

Output 9: Extract from **show macsec** output—ingress frames are being accepted

```
General MACsec Info:
  ...
  Input Data Pkts (Allowed):     257
  Input Data Pkts (Blocked):     0
```

These counters indicate what is happening to ingress traffic. The frames are either being allowed to pass through the port (after removing MACsec encapsulation) or they are blocked. If traffic is being blocked then look at other counters that show why it is being blocked.

Output when MACsec is working

Output 10: Example output from **show macsec** when MACsec is working (version 5.5.1-2)

```
awplus#show macsec interface port1.1.5

MKA and MACsec information for interface port1.1.5:

  MKA:

    Pre-shared CAK Size (Bits):    256

    Active:                        True
    Actor SCI:                     e01a.ea2a.4f49/13f1
    Actor Priority:                128
    Key Server SCI:                e01a.ea2a.4f49/13f1
    Key Server Priority:           128
    Keys Distributed:              1
    Keys Received:                 0

  Frame Generation:

    Protect Frames:                True
    Always Include SCI:            True

    Pkts Untagged:                 0
    Pkts Too Long:                 0

    Bytes Protected Only:          0
    Bytes Encrypted:               240

  Frame Verification:

    Validate Frames:               Strict
    Replay Protect:                True
    Replay Window:                 0

    Pkts Untagged (Allowed):       0
    Pkts Untagged (Blocked):       0
    Pkts Bad Tag:                  0
    Pkts Unknown SCI (Allowed):    0
    Pkts Unknown SCI (Blocked):    0
    Pkts Overrun:                  0

    Bytes Validated Only:          0
    Bytes Decrypted:               144

  Transmit Channel e01a.ea2a.4f49/13f1:

    Created Time:                  2021-11-26 17:05:55

    Pkts Protected Only:           0
    Pkts Encrypted:                5

    Secure Association 0:

      Key Identifier:              913E1BEA38E05A53120AB44500000001
      Created Time:                2021-11-26 17:06:02
      In Use:                      True
      Next Packet Number:          0x00000006

      Pkts Protected Only:         0
      Pkts Encrypted:              5
```

Output 10: Example output from **show macsec** when MACsec is working (version 5.5.1-2) (continued)

```
  Receive Channel 0800.2775.c06d/0001:

    Created Time:                  2021-11-26 17:06:00

    Pkts OK:                       3
    Pkts Unchecked:                0
    Pkts Invalid (Allowed):        0
    Pkts Invalid (Blocked):        0
    Pkts Late (Allowed):           0
    Pkts Late (Blocked):           0

    Secure Association 0:

      Key Identifier:              913E1BEA38E05A53120AB44500000001
      Created Time:                2021-11-26 17:06:02
      In Use:                      True
      Next Packet Number:          0x00000004

      Pkts OK:                     3
      Pkts Invalid (Allowed):      0
      Pkts Invalid (Blocked):      0
```

Output when there is no MACsec peer

Output 11: Example output from **show macsec** when there is no peer (version 5.5.1-2)

```
awplus#show macsec interface port1.1.5

MKA and MACsec information for interface port1.1.5:

  MKA:

    Pre-shared CAK Size (Bits):    256

    Active:                        True
    Actor SCI:                     e01a.ea2a.4f49/13f1
    Actor Priority:                128
    Key Server SCI:                -
    Key Server Priority:           -
    Keys Distributed:              0
    Keys Received:                 0

    MACsec Settings Received:      False
    MACsec Settings Accepted:      -
    MACsec Cipher Suite:           -
    MACsec Protection Mode:        -

  General MACsec Info:

    Hardware Ready:                False
    Current Cipher Suite:          -
    Cipher Suite Protection:       -

    Input Data Pkts (Allowed):     0
    Input Data Pkts (Blocked):     0
    Input EAPOL Pkts:              0

    Output Data Pkts (OK):         0
    Output Data Pkts (Error):      0
    Output EAPOL Pkts:             23
```

Output 11: Example output from **show macsec** when there is no peer (version 5.5.1-2) (continued)

```
  Frame Generation:

    Protect Frames:                 True
    Always Include SCI:             True

    Pkts Untagged:                  0
    Pkts Too Long:                  0

    Bytes Protected Only:           0
    Bytes Encrypted:                0

  Frame Verification:

    Validate Frames:                Strict
    Replay Protect:                 True
    Replay Window:                  0

    Pkts Untagged (Allowed):        0
    Pkts Untagged (Blocked):        0
    Pkts Bad Tag:                   0
    Pkts Unknown SCI (Allowed):     0
    Pkts Unknown SCI (Blocked):     0
    Pkts Overrun:                   0

    Bytes Validated Only:           0
    Bytes Decrypted:                0

  Transmit Channel e01a.ea2a.4f49/13f1:

    Created Time:                   2021-11-26 16:08:59

    Pkts Protected Only:            0
    Pkts Encrypted:                 0
```