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 3
- a description of "How MACsec Works" on page 4
- "Packet Overheads and Throughput Limits" on page 10
- information about "Configuring MACsec" on page 12, including a step by step procedure and examples.

AlliedWare Plus™ OPERATING SYSTEM

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. From version 5.4.9-2, MACsec is supported on:

- x930 (all front panel 1G ports)
- x950 (XEM2-12XS)
- SBx908 GEN2 (XEM2-12XS)

For more information, see the product's Command Reference

This documents is available from the above links on 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.

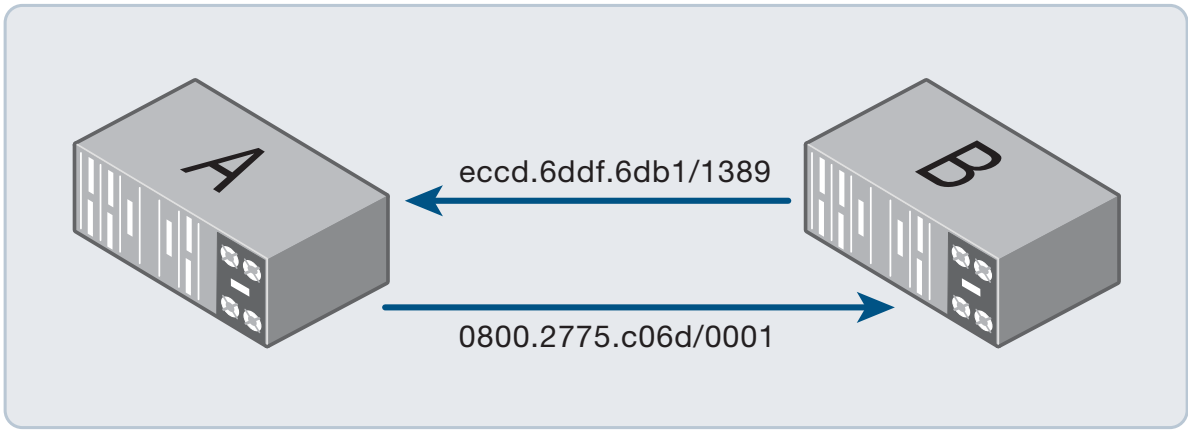
How MACsec Works

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.

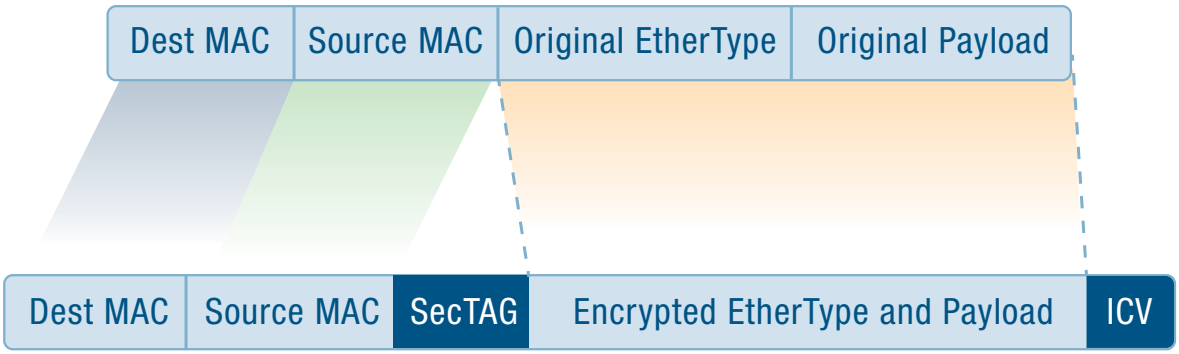
MACsec encapsulation

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

The key server:

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

AlliedWare Plus MACsec 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.

When both peers are Alliedware Plus devices, it is not important which device becomes the key server.

AlliedWare Plus supports protecting traffic with cipher suite GCM-AES-128 using integrity protection and confidentiality. If the key server chooses different MACsec protection settings then MACsec will not unblock the port

MACsec Interactions and Limitations

Supported features

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 port can only have one peer. AlliedWare Plus does not support two or more peers.
- MKA messages are always addressed to the PAE group MAC address (0180.c200.0003).
- AlliedWare Plus supports the GCM-AES-128 cipher suite for MACsec encapsulation. Other cipher suites are not supported.
- 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.

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 synchronised. During a master failover, the new stack master will:

1. reinitialize the hardware drivers of all MACsec ports so that all non-EAPOL traffic is blocked
2. wait 8 seconds for the remote MKA peer to timeout its knowledge of the old master as an MKA participant
3. restart 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.

CAK name

If you want it to be 32 bytes long then all 64 hexadecimal digits must be entered into the configuration command. AlliedWare Plus does not implicitly add zeros to the end of CAK names.

Packet Overheads and Throughput Limits

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

- "Packet overheads" on page 10—packet sizes with MACsec encapsulation
- "Throughput" on page 10—the effects of MACsec on throughput
- "Congestion on x950 and SBx908 GEN2" on page 11—how these devices deal with congestion
- "Congestion on x930" on page 11—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 5). 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 1: 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 2: 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 3: 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 3 (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 (with XEM2-12XS) switches, when traffic reaches the bandwidth limit of a MACsec-enabled port (Table 3, 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

On the x930 switch, when traffic reaches the bandwidth limits of a MACsec-enabled port on an x930 switch (Table 3, column 3), the device discards some packets at random with no priority queuing. This means that on the x930 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 to prevent network disruptions arising from congestion, see "Avoid control packet loss on x930" on page 16.

Configuring MACsec

Configuration procedure

Step 1: **Enable MACsec on the device.**

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.)

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>
```

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>
awplus(config-mka-policy)#exit
```

Step 3: **Add a pre-shared key to the interface.**

In order to authenticate each other, the MKA peer interfaces at both ends of the link must have the same pre-shared Secure Connectivity Association Key (CAK) and CAK name (CKN). The CAK must be:

- 16 bytes long, entered as 32 hexadecimal digits
- generated by a **cryptographically secure random number** generator.

We recommend using the following command on the device to generate a CAK:

```
awplus#crypto random bytes 16
```

It 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 following command.

The CAK name can be an arbitrary value from 2 to 64 hexadecimal digits long (an even number of digits, representing 1 to 32 bytes); it **must not** in any way be derived from the CAK itself, as this would seriously undermine protection. Each CAK configured on the device should have a unique name. 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 <key>
```

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: x930 only—recommended: Enable egress-rate-limiting

On the x930, you must limit traffic transmitted to prevent the loss of control frames, including MKA frames, that could otherwise interrupt traffic. See "Avoid control packet loss on x930" on page 16. You can use egress-rate-limiting.

A limit of 551700kbps is low enough for all frame-sizes. Consider your network traffic profile—you may be able to tune this for greater link utilization.

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

Step 6: x930 only—optional: Enable flow control

If flow control is feasible across the whole network, you may be able to use it to limit bandwidth instead of, or in addition to, egress limiting (above) on an x930. See "Avoid control packet loss on x930" on page 16. To enable flow control on all ports on the device, use the commands:

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

Step 7: Verify MACsec configuration.

Check the MKA policy settings.

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

The following example output shows the default policy and a user-configured policy with the key server priority set to the highest priority, and the replay window set to allow reordering of up to 64 frames.

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

```
awplus#show mka policy
MKA policy: default
  Key server priority: 128
  Replay protection: Enabled
  Replay window size: 0

MKA policy: office
  Key server priority: 0
  Replay protection: Enabled
  Replay window size: 64
```

Configuration example: MACsec with default MKA policy

This example shows the simplest configuration for SBx908 GEN2 or x950 devices at both ends of the link to protect traffic with MACsec. MACsec in this example:

- uses the default MKA policy, which has a replay window of 0 that allows for no frame reordering.
- leaves the election of the key server to be determined by the SCIs.
- x930 only: shows egress rate-limiting to limit traffic to a safe level compatible with MACsec on an x930. On an x930, remove the '!' from this command. See "Packet Overheads and Throughput Limits" on page 10.

```
...
platform macsec enable
...
```

Save the configuration and restart the device.

Use the **crypto random bytes 16** command to generate a pre-shared key, and copy this into the configuration. Note that the key entered will not be displayed in cleartext in output of the running configuration.

```
...
  interface port1.0.1
    mka pre-shared-key  ckn 01 cak <16-byte-cak>
    mka policy default
    ! Limit traffic by using egress rate limiting:
    ! egress-rate-limit 551700k
...
```

Configuration example: MACsec with modified MKA policy

This example shows the configuration for SBx908 GEN2 or x950 devices at both ends of the link to protect traffic with MACsec. This example:

- Uses new MKA policy ‘office’, which allows you to modify the default settings.
- Optionally increases the window size for replay protection.
- Optionally provides different key server priorities at the two ends of the link.
- x930 only: Shows egress rate-limiting to limit traffic to a safe level compatible with MACsec on the x930. On an x930, remove the ‘!’ from this command. See "Packet Overheads and Throughput Limits" on page 10.

These options have been commented out in the extract below. To use the configuration lines that have been commented, remove the ‘!’.

```
...
platform macsec enable
...
```

Save the configuration and restart the device.

Use the **crypto random bytes 16** command to generate a pre-shared key, and copy this into the configuration. Note that the key entered will not be displayed in cleartext in output of the running configuration.

```
...
  mka policy office
    ! Increase the replay protection window to 64 frames:
    ! macsec replay-protection window-size 64
    ! Set this device to the highest priority to become the MACsec key server:
    ! key-server priority 0
  interface port1.0.1
    mka pre-shared-key  ckn 01 cak <16-byte-cak>
    mka policy office
    ! Limit traffic by using egress rate limiting:
    ! egress-rate-limit 551700k
...
```

Avoid control packet loss on x930

On an x930, 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 16 on MACsec-enabled ports on any x930 to limit bandwidth and prevent control packet loss by MACsec. This may limit bandwidth more than necessary.
- "Calculating an egress-rate-limit" on page 17 may allow you to tune the limit for bandwidth utilisation.
- "Flow control" on page 18 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.

For the commands used to configure the solution, see the "Configuration procedure" on page 12. For more information about why this is necessary, see "Packet Overheads and Throughput Limits" on page 10.

Egress-rate-limiting

You can use egress-rate-limiting on the MACsec port to limit traffic to a safe level on any 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 3 on page 11, column 3). You can choose a limit (551700k) that is low enough to ensure packets are never randomly discarded regardless of frame-size, even if all or most frames are 64 bytes.

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 1000 Mbps link with a 5% safety buffer for frames of average 'size' in kB. This is likely to avoid dropping important packets.

```
(frame-size / (frame-size + 52))k x 1 000 000 x 0.95 = (egress-rate-limit)k
```

For example, if you determine that the average frame-size is 800 bytes, calculate an appropriate egress-rate-limit value by adding 52 bytes and dividing the original value by this, then multiplying by 1 000 000 and subtracting a safety buffer of 5%:

```
(800 / 852)k x 1000000 x 0.95 = 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

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.

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

This section provides:

- "Example output from show macsec when MACsec is working" on page 19
- "Example output from show macsec when there is no peer" on page 21
- Information about "Selected parameters in the output from ‘show macsec’" on page 22

Output 3: Example output from **show macsec** when MACsec is working

```
awplus#show macsec interface port1.0.1

MKA and MACsec information for interface port1.0.1:

  MKA:

    Active:                         True
    MACsec Protection Mode:         Integrity and confidentiality
    Actor SCI:                      eccd.6ddf.6db1/1389
    Actor Priority:                 128
    Key Server SCI:                 eccd.6ddf.6db1/1389
    Key Server Priority:            128
    Keys Distributed:               1
    Keys Received:                  0

  General MACsec Info:

    State:                          OK
    Current Cipher Suite:           GCM-AES-128
    Cipher Suite Protection:        Integrity and confidentiality

    Input Data Pkts (Allowed):      3
    Input Data Pkts (Blocked):      0
    Input EAPOL Pkts:               2

    Output Data Pkts (OK):          5
    Output Data Pkts (Error):       0
    Output EAPOL Pkts:              26

  Frame Generation:

    Protect Frames:                 True
    Always Include SCI:             True

    Pkts Untagged:                  0
    Pkts Too Long:                  0

    Bytes Protected Only:           0
    Bytes Encrypted:                240
```

Output 3: Example output from **show macsec** when MACsec is working (continued)

```
  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:                246

  Transmit Channel eccd.6ddf.6db1/1389:

    Created Time:                   2019-11-07 22:16:55

    Pkts Protected Only:            0
    Pkts Encrypted:                 5

    Secure Association 0:

      Key Identifier:               ABA7E9B021C4A546865757AE00000001
      Created Time:                 2019-11-07 22:17:45
      In Use:                       True
      Next Packet Number:           0x00000006

      Pkts Protected Only:          0
      Pkts Encrypted:               5

  Receive Channel 0800.2775.c06d/0001:

    Created Time:                   2019-11-07 22:17:43

    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:               ABA7E9B021C4A546865757AE00000001
      Created Time:                 2019-11-07 22:17:45
      In Use:                       True
      Next Packet Number:           0x00000004

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

Output 4: Example output from **show macsec** when there is no peer

```
awplus#show macsec interface port1.0.1
MKA and MACsec information for interface port1.0.1:

  MKA:

    Active:                         True
    MACsec Protection Mode:         Undecided
    Actor SCI:                      eccd.6ddf.6db1/1389
    Actor Priority:                 128
    Key Server SCI:                 -
    Key Server Priority:            -
    Keys Distributed:               0
    Keys Received:                  0

  General MACsec Info:

    State:                          Bad
    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:              22

  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 eccd.6ddf.6db1/1389:

    Created Time:                   2019-11-07 22:16:55

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

Selected parameters in the output from ‘show macsec’

MKA:

- **MACsec Protection Mode**
  The MACsec protection mode that the key server has chosen. Only ‘Integrity and confidentiality’ is supported by AlliedWare Plus. This will show ‘Undecided’ when a key server has not been elected.

General MACsec Information:

- **State**
  This displays ‘OK’ when the necessary settings have been applied to hardware. That is, the channels and SAs have been added and it is using the cipher suite and MACsec protection mode that AlliedWare Plus supports. Otherwise, it displays ‘Bad’ and MACsec protection is not working.
- **Input EAPOL Pkts**
  This counts the number of ingress EAPOL frames (including MKA) that have been allowed through the port without MACsec protection. If this is not going up then the device is not receiving MKA messages from the peer. See the **show auth statistics interface** command for a counter that includes only MKA messages.
- **Input Data Pkts**
  These two counters indicate what is happening to other kinds of ingress traffic—either they are 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.

Frame Verification:

- **Pkts Untagged (Blocked)**
  This goes up when ingress traffic does not have MACsec encapsulation.

Receive Channel

- **Pkts Late (Blocked)**
  This goes up when frames are being blocked by replay protection.

Allied Telesis NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com