

# Probleemoplossing voor MACSEC op Catalyst 9000

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Voordelen van MacSec](#)

[MACsec en MTU](#)

[Waar MACsec wordt gebruikt](#)

[Terminologie](#)

[Scenario 1: MACsec Switch-naar-Switch koppelt beveiliging met SAP in Pre-Shared Key \(PSK\) modus](#)

[Topologie](#)

[Scenario 2: MACsec Switch-to-Switch Link Security met MKA in vooraf gedeelde sleutel \(PSK\) modus](#)

[Topologie](#)

[Voorbeeld van opvulling](#)

[Andere configuratieopties](#)

[MACsec Switch-to-Switch Link-beveiliging met MKA op gebundelde/poortkanaals interface](#)

[MACsec Switch-to-Switch Link Security via L2 tussenliggende switches, PSK-modus](#)

[Beperkingen](#)

[Operationele informatie over MACsec](#)

[Bedieningssequentie](#)

[MACsec-pakketten](#)

[SAP-onderhandeling](#)

[Key exchange](#)

[MACsec op platform](#)

[Productcompatibiliteitstabel](#)

[Gerelateerde informatie](#)

## Inleiding

Dit document beschrijft de functie MACsec, de gebruikscases en hoe u problemen kunt oplossen met de functie op Catalyst 9000 switches. Het bereik van dit document is MACsec op LAN, tussen twee switches/routers.

## Voorwaarden

### Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

### Gebruikte componenten

- C9300
- C9400
- C9500

- C9600

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

---

**Opmerking:** raadpleeg de juiste configuratiehandleiding voor de opdrachten die worden gebruikt om deze functies op andere Cisco-platforms in te schakelen.

---

## Achtergrondinformatie

De duidelijke communicatie van tekstgegevens is vatbaar voor veiligheidsbedreigingen. De inbreuken van de veiligheid kunnen op om het even welke laag van het OSI model voorkomen. Enkele van de veelvoorkomende inbreuken op Layer 2 zijn snuffelen, pakketafluisteren, geknoei, injectie, MAC-adresspoofing, ARP-spoofing, Denial of Service (DoS)-aanvallen tegen een DHCP-server en VLAN-hopping.

MacSec is een L2-coderingstechnologie die wordt beschreven in de IEEE 802.1AE-standaard. MACsec beveiligt de gegevens op fysieke media en maakt het onmogelijk dat gegevens op hogere lagen worden gecompromitteerd. Hierdoor krijgt de MACsec-codering voorrang op elke andere coderingsmethode voor hogere lagen, zoals IPsec en SSL.

### Voordelen van MacSec

**Clientgeoriënteerde modus:** MACsec wordt gebruikt in instellingen waarin twee switches die met elkaar samenwerken als een sleutelservers of een sleutelclient kunnen wisselen voordat sleutels worden uitgewisseld. De toetsserver genereert en onderhoudt de CAK tussen de twee peers.

**Data Integrity Check:** MACsec gebruikt MKA om een Integrity Check Value (ICV) te genereren voor het frame dat op de poort aankomt. Als de gegenereerde ICV hetzelfde is als de ICV in het frame, wordt het frame geaccepteerd; anders wordt het verwijderd.

**Gegevensversleuteling:** MACsec biedt poortcodering op de interfaces van switches. Dit betekent dat de frames die uit de geconfigureerde poort worden verzonden, worden versleuteld en dat de frames die op de poort worden ontvangen, worden gedecrypteerd. MACsec biedt ook een mechanisme waar u kunt configureren of alleen versleutelde frames of alle

de frames (versleuteld en gewoon) worden op de interface geaccepteerd.

**Replay Protection:** Wanneer frames worden verzonden via het netwerk, is er een mogelijkheid dat frames uit de volgorde komen. MACsec biedt een configureerbaar venster dat een gespecificeerd aantal out-of-sequentiefames accepteert.

### MACsec en MTU

De MACsec-header wordt aangevuld met 32 bytes header-overhead. Overweeg een grotere systeem/interface MTU op switches in het pad om rekening te houden met de extra overhead toegevoegd door de MACsec header. Als MTU te laag is, kunt u onverwacht pakketverlies/vertraging zien voor toepassingen die hogere MTU moeten gebruiken.

---

**Opmerking:** als er een probleem is met betrekking tot MACSEC, zorg er dan voor dat de GBIC aan beide uiteinden wordt ondersteund volgens de [Compatibiliteitsmatrix](#) .

---

## Waar MACsec wordt gebruikt

### Campus Use Cases

- Host-to-switch
- Tussen locaties of gebouwen
- Tussen vloeren in een multi-tenancy

### Gebruikscases voor datacenters

- Interconnect voor datacenters
- Server-naar-switch

### WAN-gebruikscases

- Interconnect voor datacenters
- Campus interconnect
- hubspraak

## Terminologie

<b>MKA</b>	MACsec-sleutelovereenkomst	gedefinieerd in IEEE 802.1X REV-2010 als een sleutelprotocol voor het ontdekken van MACsec-peers en onderhandelingsleutels
<b>CAK</b>	Sleutel voor connectiviteitsassociatie	lange-leven hoofdsleutel gebruikt om alle andere sleutels te produceren die voor MACsec worden gebruikt. LAN-implementaties halen dit uit MSK (gegenereerd tijdens EAP-uitwisseling)
<b>PMK</b>	Pairwise Master Key	Een van de componenten die worden gebruikt om de sessiesleutels af te leiden die worden gebruikt om verkeer te versleutelen. Handmatig geconfigureerd of afgeleid van 802.1X
<b>CKN</b>	CAK-sleutelnaam	gebruikt om de toetswaarde of CAK te configureren. Zelfs het aantal <u>HEX-tekens</u> tot 64 tekens is toegestaan.
<b>SAK</b>	Secure-associatiesleutel	afgeleid door de gekozen Key Server van de CAK en is de sleutel die door de router/eindapparaten wordt gebruikt om verkeer voor een bepaalde sessie te versleutelen.
<b>ICV</b>	Integriteitscontrole - Value-toets	afgeleid van CAK en is gelabeld in elk data/control frame om te bewijzen dat het frame van een erkende peer is. 8-16 bytes, afhankelijk van de suite
<b>KEK</b>	Key versleuteling	afgeleid van CAK (de vooraf gedeelde sleutel) en gebruikt om de MacSec-toetsen te beschermen
<b>SCI</b>	Identificatiecode voor beveiligde kanalen	Elke virtuele poort ontvangt een unieke beveiligde channel identifier (SCI), gebaseerd op het MAC-adres van de fysieke interface aaneengeschakeld met een 16-bits poort-ID

## Scenario 1: MACsec Switch-naar-Switch koppelt beveiliging met SAP in Pre-Shared Key (PSK) modus

### Topologie



```
mode-list gcm-encrypt
```

**NOTE:**

```
cts manual
```

```
<-- Supplies local configuration for Cisco TrustSec parameters
```

```
no propagate sgt
```

```
<-- disable SGT tagging on a manually-configured TrustSec-capable interface,
```

```
if you do not need to propage the SGT tags.
```

```
sap pmk AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA mode-list gcm-encrypt
```

```
<--
```

Use the sap command to manually specify the Pairwise Master Key (PMK) and the Security Association Protocol

authentication and encryption modes to negotiate MACsec link encryption between two interfaces.

The default encryption is sap modelist gcm-encrypt null

```
9300_stack#(config-if-cts-manual)#
```

```
sap pmk fa mode-list
```

```
?
```

```
gcm-encrypt GCM authentication, GCM encryption
```

```
gmac GCM authentication, no encryption
```

```
no-encap No encapsulation
```

```
null Encapsulation present, no authentication, no encryption
```

Use "gcm-encrypt" for full GCM-AES-128 encryption.

These protection levels are supported when you configure SAP pairwise master key (sap pmk):

SAP is not configuredâ€”no protection.

sap mode-list gcm-encrypt gmac no-encapâ€”protection desirable but not mandatory.

sap mode-list gcm-encrypt gmacâ€”confidentiality preferred and integrity required.

The protection is selected by the supplicant according to supplicant preference.

sap mode-list gmac â€”integrity only.

sap mode-list gcm-encrypt-confidentiality required.  
sap mode-list gmac gcm-encrypt-integrity required and preferred, confidentiality optional.

## Stap 2. Controleer de MACsec-status en de parameters/tellers zijn juist

```
<#root>
```

```
### Ping issued between endpoints to demonstrate counters ###
```

```
Host-1#
```

```
ping 10.10.10.12 <-- sourced from Host-1 IP 10.10.10.11
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
9300_stack#
```

```
sh macsec summary
```

```
Interface
```

```
Transmit SC      Receive SC <-- Secure Channel (SC) flag is set for transmit and receive
```

```
GigabitEthernet1/0/1
```

```
1                1
```

```
9300_stack#
```

```
sh macsec interface gigabitEthernet 1/0/1
```

```
MACsec is enabled
```

```
Replay protect : enabled  
Replay window : 0  
Include SCI : yes  
Use ES Enable : no  
Use SCB Enable : no  
Admin Pt2Pt MAC : forceTrue(1)  
Pt2Pt MAC Operational : no  
  
Cipher : GCM-AES-128
```

```
Confidentiality Offset : 0  
!
```

```
Capabilities
```

```
ICV length : 16
```

Data length change supported: yes  
Max. Rx SA : 16  
Max. Tx SA : 16  
Max. Rx SC : 8  
Max. Tx SC : 8  
Validate Frames : strict  
PN threshold notification support : Yes

Ciphers supported :

GCM-AES-128

GCM-AES-256

GCM-AES-XPB-128

GCM-AES-XPB-256

!

Transmit Secure Channels

SCI : 682C7B9A4D010000  
SC state : notInUse(2)

Elapsed time : 03:17:50

Start time : 7w0d  
Current AN: 0  
Previous AN: 1  
Next PN: 185  
SA State: notInUse(2)  
Confidentiality : yes  
SAK Unchanged : no

SA Create time : 03:58:39

SA Start time : 7w0d

SC Statistics  
Auth-only Pkts : 0  
Auth-only Bytes : 0

Encrypt Pkts : 2077

Encrypt Bytes : 0

!

SA Statistics

Auth-only Pkts : 0

Encrypt Pkts : 184

<-- packets are being encrypted and transmitted on this link

!  
Port Statistics  
Egress untag pkts 0  
Egress long pkts 0  
!

Receive Secure Channels

SCI : D0C78970C3810000  
SC state : notInUse(2)  
Elapsed time : 03:17:50  
Start time : 7w0d  
Current AN: 0  
Previous AN: 1  
Next PN: 2503  
RX SA Count: 0  
SA State: notInUse(2)  
SAK Unchanged : no

SA Create time : 03:58:39

SA Start time : 7w0d

SC Statistics  
Notvalid pkts 0  
Invalid pkts 0  
Valid pkts 28312  
Valid bytes 0  
Late pkts 0  
Uncheck pkts 0  
Delay pkts 0  
UnusedSA pkts 0  
NousingSA pkts 0  
Decrypt bytes 0

!  
SA Statistics

Notvalid pkts 0  
Invalid pkts 0

Valid pkts 2502

<-- number of valid packets received on this link

UnusedSA pkts 0



NousingSA pkts 0

!

Port Statistics

Ingress untag pkts 0

Ingress notag pkts 36

Ingress badtag pkts 0

Ingress unknownSCI pkts 0

Ingress noSCI pkts 0

Ingress overrun pkts 0

!

9300\_stack#

sh cts interface summary

Global Dot1x feature is Disabled

CTS Layer2 Interfaces

-----

Interface	Mode	IFC-state	dot1x-role	peer-id	IFC-cache	Critical-Authentication
-----------	------	-----------	------------	---------	-----------	-------------------------

-----

Gi1/0/1

MANUAL OPEN

unknown	unknown	invalid	Invalid
---------	---------	---------	---------

CTS Layer3 Interfaces

-----

Interface	IPv4 encap	IPv6 encap	IPv4 policy	IPv6 policy
-----------	------------	------------	-------------	-------------

-----

!

9300\_stack#

sh cts interface gigabitEthernet 1/0/1

Global Dot1x feature is Disabled

Interface GigabitEthernet1/0/1:

CTS is enabled, mode: MANUAL

IFC state: OPEN

Interface Active for 04:10:15.723 <--- Uptime of MACsec port

Authentication Status: NOT APPLICABLE

Peer identity: "unknown"

Peer's advertised capabilities: "sap"

Authorization Status: NOT APPLICABLE

!

SAP Status: SUCCEEDED <-- SAP is successful

Version: 2

Configured pairwise ciphers:

gcm-encrypt

!

Replay protection: enabled

Replay protection mode: STRICT

!  
Selected cipher: gcm-encrypt  
!  
Propagate SGT: Disabled  
Cache Info:  
Expiration : N/A  
Cache applied to link : NONE  
!  
Statistics:  
  authc success: 0  
  authc reject: 0  
  authc failure: 0  
  authc no response: 0  
  authc logoff: 0

sap success: 1 <-- Negotiated once

sap fail: 0 <-- No failures

  authz success: 0

  authz fail: 0

port auth fail: 0

L3 IPM: disabled

**Stap 3.** De software van het overzicht zuivert wanneer de verbinding omhoog komt.

<#root>

### Verify CTS and SAP events ###

debug cts sap events  
debug cts sap packets

### Troubleshoot MKA session bring up issues ###

debug mka event  
debug mka errors  
debug mka packets

### Troubleshoot MKA keep-alive issues ###

debug mka linksec-interface  
debug mka macsec  
debug macsec

\*May 8 00:48:04.843: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to down  
\*May 8 00:48:05.324: Macsec interface GigabitEthernet1/0/1 is UP

\*May 8 00:48:05.324: CTS SAP ev (Gi1/0/1): Session started (new).

\*May 8 00:48:05.324: cts\_sap\_session\_start CTS SAP ev (Gi1/0/1) peer:0000.0000.0000 AAAAAAAAAAAAAAAAAAAAAA

CTS SAP ev (Gi1/0/1): Old state: [waiting to restart],  
event: [restart timer expired], action:

[send message #0] succeeded.

New state: [waiting to receive message #1].

\*May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381 <-- MAC of peer switch

\*May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): EAPOL-Key message #0 parsed and validated.

\*May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): Our MAC = 682C.7B9A.4D01 <-- MAC of local interface

peer's MAC = D0C7.8970.C381.

CTS SAP ev (Gi1/0/1): Old state: [waiting to receive message #1],

event: [received message #0], action: [break tie] succeeded.

New state: [determining role].

\*May 8 00:48:05.449: cts\_sap\_generate\_pmkid\_and\_sci CTS SAP ev (Gi1/0/1) auth:682c.7b9a.4d01 supp:d0c7.8970.c381

CTS SAP ev (Gi1/0/1): Old state: [determining role],

event: [change to authenticator], action: [send message #1] succeeded.

New state: [waiting to receive message #2].

\*May 8 00:48:05.457: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381.

CTS SAP ev (Gi1/0/1): New keys derived:

KCK = 700BEF1D 7A8E10F7 1243A168 883C74FB,

KEK = C207177C B6091790 F3C5B4B1 D51B75B8,

TK = 1B0E17CD 420D12AE 7DE06941 B679ED22,

\*May 8 00:48:05.457: CTS SAP ev (Gi1/0/1): EAPOL-Key message #2 parsed and validated.

\*May 8 00:48:05.457: CTS-SAP ev: cts\_sap\_action\_program\_msg\_2: (Gi1/0/1) GCM is allowed.

\*May 8 00:48:05.457: MACSec-IPC: sending clear\_frames\_option  
\*May 8 00:48:05.457: MACSec-IPC: geting switch number  
\*May 8 00:48:05.457: MACSec-IPC: switch number is 1  
\*May 8 00:48:05.457: MACSec-IPC: clear\_frame send msg success  
\*May 8 00:48:05.457: MACSec-IPC: getting macsec clear frames response  
\*May 8 00:48:05.457: MACSec-IPC: watched boolean waken up  
\*May 8 00:48:05.457: MACsec-CTS: create\_sa invoked for SA creation  
\*May 8 00:48:05.457: MACsec-CTS: Set up TxSC and RxSC before we installTxSA and RxSA  
\*May 8 00:48:05.457: MACsec-CTS: create\_tx\_sc, avail=yes sci=682C7B9A  
\*May 8 00:48:05.457: NGWC-MACSec: create\_tx\_sc vlan invalid  
\*May 8 00:48:05.457: NGWC-MACSec: create\_tx\_sc client vlan=1, sci=0x682C7B9A4D010000  
\*May 8 00:48:05.457: MACSec-IPC: sending create\_tx\_sc  
\*May 8 00:48:05.457: MACSec-IPC: geting switch number  
\*May 8 00:48:05.457: MACSec-IPC: switch number is 1  
\*May 8 00:48:05.457: MACSec-IPC: create\_tx\_sc send msg success  
\*May 8 00:48:05.458: MACsec API blocking the invoking context  
\*May 8 00:48:05.458: MACSec-IPC: getting macsec sa\_sc response  
\*May 8 00:48:05.458: macsec\_blocking\_callback  
\*May 8 00:48:05.458: Wake up the blocking process  
\*May 8 00:48:05.458: MACsec-CTS: create\_rx\_sc, avail=yes sci=D0C78970  
\*May 8 00:48:05.458: NGWC-MACSec: create\_rx\_sc client vlan=1, sci=0xD0C78970C3810000  
\*May 8 00:48:05.458: MACSec-IPC: sending create\_rx\_sc  
\*May 8 00:48:05.458: MACSec-IPC: geting switch number  
\*May 8 00:48:05.458: MACSec-IPC: switch number is 1  
\*May 8 00:48:05.458: MACSec-IPC: create\_rx\_sc send msg success  
\*May 8 00:48:05.458: MACsec API blocking the invoking context  
\*May 8 00:48:05.458: MACSec-IPC: getting macsec sa\_sc response  
\*May 8 00:48:05.458: macsec\_blocking\_callback  
\*May 8 00:48:05.458: Wake up the blocking process  
\*May 8 00:48:05.458: MACsec-CTS: create\_tx\_rx\_sa, txsci=682C7B9A, an=0  
\*May 8 00:48:05.458: MACSec-IPC: sending install\_tx\_sa  
\*May 8 00:48:05.458: MACSec-IPC: geting switch number  
\*May 8 00:48:05.458: MACSec-IPC: switch number is 1  
\*May 8 00:48:05.459: MACSec-IPC: install\_tx\_sa send msg success  
\*May 8 00:48:05.459: NGWC-MACSec:Sending authorized event to port SM  
\*May 8 00:48:05.459: MACsec API blocking the invoking context  
\*May 8 00:48:05.459: MACSec-IPC: getting macsec sa\_sc response  
\*May 8 00:48:05.459: macsec\_blocking\_callback  
\*May 8 00:48:05.459: Wake up the blocking process  
\*May 8 00:48:05.459: MACsec-CTS: create\_tx\_rx\_sa, rxsci=D0C78970, an=0  
\*May 8 00:48:05.459: MACSec-IPC: sending install\_rx\_sa  
\*May 8 00:48:05.459: MACSec-IPC: geting switch number  
\*May 8 00:48:05.459: MACSec-IPC: switch number is 1  
\*May 8 00:48:05.460: MACSec-IPC: install\_rx\_sa send msg success  
\*May 8 00:48:05.460: MACsec API blocking the invoking context  
\*May 8 00:48:05.460: MACSec-IPC: getting macsec sa\_sc response  
\*May 8 00:48:05.460: macsec\_blocking\_callback  
\*May 8 00:48:05.460: Wake up the blocking process

CTS SAP ev (Gi1/0/1): Old state: [waiting to receive message #2],  
event: [received message #2], action: [program message #2] succeeded.  
New state: [waiting to program message #2].

CTS SAP ev (Gi1/0/1): Old state: [waiting to program message #2],  
event: [data path programmed], action: [send message #3] succeeded.

New state: [waiting to receive message #4].

```
*May 8 00:48:05.467: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381.

*May 8 00:48:05.467: CTS SAP ev (Gi1/0/1): EAPOL-Key message #4 parsed and validated.

*May 8 00:48:05.473: CTS-SAP ev: cts_sap_sync_sap_info: incr sync msg sent for Gi1/0/1

*May 8 00:48:07.324: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
```

#### Stap 4. Beoordeel Platform level traces wanneer de link omhoog komt

```
<#root>
```

```
9300_stack#
```

```
sh platform software fed switch 1 ifm mappings
```

Interface	IF_ID	Inst	Asic	Core	Port	SubPort	Mac	Cntx	LPN	GPN	Type	Active
GigabitEthernet1/0/1	0x8	1	0	1	0	0	26	6	1	1	NIF	Y

Note the IF\_ID for respective intf

- This respective IF\_ID shows in MACSEC FED traces seen here.

```
9300_stack#
```

```
set platform software trace fed switch 1 cts_aci verbose
```

```
9300_stack#
```

```
set platform software trace fed switch 1 macsec verbose
```

```
<-- switch number with MACsec port
```

```
9300_stack#
```

```
request platform software trace rotate all
```

/// shut/no shut the MACsec interface ///

9300\_stack#

show platform software trace message fed switch 1

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec\_

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Running Install

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install RxSA ca

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec install F

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering ins\_rx\_

2019/05/08 01:08:50.688 {fed\_F0-0}{1}: [l2tunnel\_bcast] [16837]: UUID: 0, ra: 0, TID: 0 (ERR): port\_id 0

2019/05/08 01:08:50.687 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec\_

2019/05/08 01:08:50.687 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.687 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if\_id = 8, cts\_

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Calling Install

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create time of

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install TxSA ca

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec install T

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering ins\_tx

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Conf\_Offset in

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Successfully in

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Secy policy har

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install policy

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Attach policy

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Creating drop e

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if\_id = 8, cts

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create RxSC cal

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec create RX

2019/05/08 01:08:50.686 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering cre\_rx

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): txSC setting xp

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Conf\_Offset in

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if\_id = 8, cts

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): secy created su

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if\_id = 8, cts

2019/05/08 01:08:50.685 {fed\_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if\_id = 8, cts

```

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): is_remote is 0

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create TxSC call

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec create TX
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering cre_tx
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent clear_f
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec clear_fra
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [macsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering clear_f
2019/05/08 01:08:50.527 {fed_F0-0}{1}: [pm_xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): XCVR POST:XCVR
2019/05/08 01:08:50.525 {fed_F0-0}{1}: [xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): ntfy_lnk_status: M

2019/05/08 01:08:48.142 {fed_F0-0}{1}: [pm_xcvr] [16837]: UUID: 0, ra: 0, TID: 0 (note): Enable XCVR for

2019/05/08 01:08:48.142 {fed_F0-0}{1}: [pm_tdl] [16837]: UUID: 0, ra: 0, TID: 0 (note): Received PM port

```

## Step 5. Controleer de status van de MACsec-interface in de hardware

```
<#root>
```

```
9300_stack#
```

```
sh platform pm interface-numbers
```

```
interface iif-id gid slot unit slun HWIDB-Ptr status status2 state snmp-if-index
```

```
-----
```

interface	iif-id	gid	slot	unit	slun	HWIDB-Ptr	status	status2	state	snmp-if-index
Gil/0/1	8	1	1	1	1	0x7F2C90D7C600	0x10040	0x20001B	0x4	8

```
9300_stack#
```

```
sh pl software fed switch 1 ifm if-id 8 <-- iif-id 8 maps to gig1/0/1
```

```
Interface IF_ID : 0x0000000000000008
```

```
Interface Name : GigabitEthernet1/0/1
```

```
Interface Block Pointer : 0x7f4a6c66b1b8
```

```
Interface Block State : READY
```

```
Interface State : Enabled
```

```
Interface Status : ADD, UPD
```

```
Interface Ref-Cnt : 8
```

```
Interface Type : ETHER
```



Port Type : SWITCH PORT

Port Location : LOCAL

Slot : 1  
Unit : 0  
Slot Unit : 1  
SNMP IF Index : 8  
GPN : 1  
EC Channel : 0  
EC Index : 0

Port Handle : 0x4e00004c

LISP v4 Mobility : false  
LISP v6 Mobility : false  
QoS Trust Type : 3  
!

Port Information

Handle ..... [0x4e00004c]  
Type ..... [Layer2]  
Identifier ..... [0x8]  
Slot ..... [1]  
Unit ..... [1]

Port Physical Subblock

Affinity ..... [local]  
Asic Instance ..... [1 (A:0,C:1)]  
AsicPort ..... [0]  
AsicSubPort ..... [0]  
MacNum ..... [26]  
ContextId ..... [6]  
LPN ..... [1]  
GPN ..... [1]  
Speed ..... [1GB]  
type ..... [NIF]  
  
PORT\_LE ..... [0x7f4a6c676bc8]

<--- port\_LE

L3IF\_LE ..... [0x0]  
DI ..... [0x7f4a6c67d718]  
SubIf count ..... [0]

Port L2 Subblock

Enabled ..... [Yes]  
Allow dot1q ..... [Yes]  
Allow native ..... [Yes]  
Default VLAN ..... [1]  
Allow priority tag ... [Yes]  
Allow unknown unicast [Yes]  
Allow unknown multicast[Yes]  
Allow unknown broadcast[Yes]  
Allow unknown multicast[Enabled]  
Allow unknown unicast [Enabled]  
Protected ..... [No]  
IPv4 ARP snoop ..... [No]  
IPv6 ARP snoop ..... [No]

Jumbo MTU ..... [1500]  
Learning Mode ..... [1]  
Vepa ..... [Disabled]

Port QoS Subblock

Trust Type ..... [0x2]  
Default Value ..... [0]  
Ingress Table Map ..... [0x0]  
Egress Table Map ..... [0x0]  
Queue Map ..... [0x0]

Port Netflow Subblock

Port Policy Subblock

List of Ingress Policies attached to an interface

List of Egress Policies attached to an interface

Port CTS Subblock

Disable SGACL ..... [0x0]  
Trust ..... [0x0]  
Propagate ..... [0x0]  
%Port SGT ..... [-1717360783]

Physical Port Macsec Subblock <-- This block is not present when MACSEC is not enabled

Macsec Enable .... [Yes]

Macsec port handle.... [0x4e00004c] <-- Same as PORT\_LE

Macsec Virtual port handles....

.....[0x11000005]

Macsec Rx start index.... [0]  
Macsec Rx end index.... [6]  
Macsec Tx start index.... [0]  
Macsec Tx end index.... [6]

Ref Count : 8 (feature Ref Counts + 1)

IFM Feature Ref Counts

FID : 102 (AAL\_FEATURE\_SRTP), Ref Count : 1  
FID : 59 (AAL\_FEATURE\_NETFLOW\_ACL), Ref Count : 1  
FID : 95 (AAL\_FEATURE\_L2\_MULTICAST\_IGMP), Ref Count : 1  
FID : 119 (AAL\_FEATURE\_PV\_HASH), Ref Count : 1  
FID : 17 (AAL\_FEATURE\_PBB), Ref Count : 1  
FID : 83 (AAL\_FEATURE\_L2\_MATM), Ref Count : 1  
FID : 30 (AAL\_FEATURE\_URPF\_ACL), Ref Count : 1

IFM Feature Sub block information

FID : 102 (AAL\_FEATURE\_SRTP), Private Data : 0x7f4a6c9a0838  
FID : 59 (AAL\_FEATURE\_NETFLOW\_ACL), Private Data : 0x7f4a6c9a00f8  
FID : 17 (AAL\_FEATURE\_PBB), Private Data : 0x7f4a6c9986b8  
FID : 30 (AAL\_FEATURE\_URPF\_ACL), Private Data : 0x7f4a6c9981c8

9300\_stack#

```
sh pl hard fed switch 1 fwd-asic abstraction print-resource-handle 0x7f4a6c676bc8 1 <-- port_LE handle
```

```
Handle:0x7f4a6c676bc8 Res-Type:ASIC_RSC_PORT_LE Res-Switch-Num:0 Asic-Num:1 Feature-ID:AL_FID_IFM Lkp-ft
priv_ri/priv_si Handle: (nil)Hardware Indices/Handles: index1:0x0 mtu_index/l3u_ri_index1:0x2 sm handle
Detailed Resource Information (ASIC# 1)
```

\*\*snip\*\*

```
LEAD_PORT_ALLOW_CTS value 0 Pass
```

```
LEAD_PORT_ALLOW_NON_CTS value 0 Pass
```

```
LEAD_PORT_CTS_ENABLED value 1 Pass <-- Flag = 1 (CTS enabled)
```

```
LEAD_PORT_MACSEC_ENCRYPTED value 1 Pass <-- Flag = 1 (MACsec encrypt enabled)
```

```
LEAD_PORT_PHY_MAC_SEC_SUB_PORT_ENABLED value 0 Pass
```

```
LEAD_PORT_SGT_ALLOWED value 0 Pass
```

```
LEAD_PORT_EGRESS_MAC_SEC_ENABLE_WITH_SCI value 1 Pass <-- Flag = 1 (MACsec with SCI enabled)
```

```
LEAD_PORT_EGRESS_MAC_SEC_ENABLE_WITHOUT_SCI value 0 Pass
```

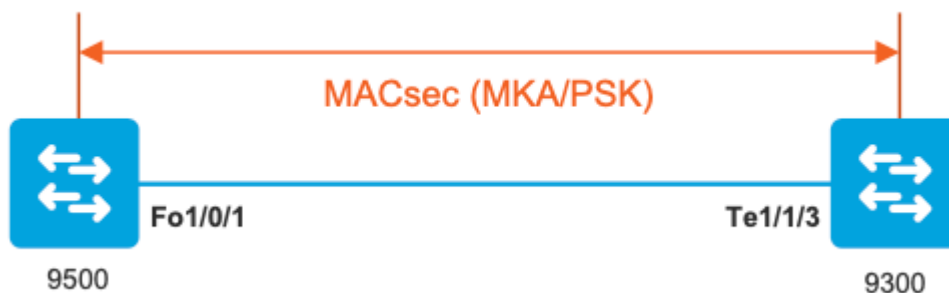
```
LEAD_PORT_EGRESS_MAC_SEC_SUB_PORT value 0 Pass
```

```
LEAD_PORT_EGRESS_MACSEC_ENCRYPTED value 0 Pass
```

\*\*snip\*\*

## Scenario 2: MACsec Switch-to-Switch Link Security met MKA in vooraf gedeelde sleutel (PSK) modus

### Topologie



### Stap 1. Valideren van de configuratie aan beide zijden van de link

```
<#root>
```

```
C9500#
```

```
sh run | sec key chain
```

```
key chain KEY macsec
key 01
  cryptographic-algorithm aes-256-cmac
  key-string 7 101C0B1A0343475954532E2E767B3233214105150555030A0004500B514B175F5B05515153005E0E5E505C525
  lifetime local 00:00:00 Aug 21 2019 infinite <-- use NTP to sync the time for key chains
```

```
mka policy MKA
```

```
key-server priority 200
macsec-cipher-suite gcm-aes-256
confidentiality-offset 0
```

```
C9500#
```

```
sh run interface fo1/0/1
```

```
interface fo1/0/1
macsec network-link
```

```
mka policy MKA
```

```
mka pre-shared-key key-chain KEY
```

```
C9300#
```

```
sh run interface tel1/1/3
```

```
interface tel1/1/3
macsec network-link
```

```
mka policy MKA
```

```
mka pre-shared-key key-chain KEY
```

**Stap 2.** Validate MACsec is ingeschakeld en alle parameters/tellers zijn correct

```
<#root>
```

```
### This example shows the output from one side, verify on both ends of MACSEC tunnel ###
```

```
C9500#
```

```
sh macsec summary
```

Interface	Transmit SC	Receive SC
FortyGigabitEthernet1/0/1	1	1

C9500#

sh macsec interface fortyGigabitEthernet 1/0/1

MACsec is enabled

Replay protect : enabled  
Replay window : 0  
Include SCI : yes  
Use ES Enable : no  
Use SCB Enable : no  
Admin Pt2Pt MAC : forceTrue(1)  
Pt2Pt MAC Operational : no  
Cipher : GCM-AES-256

Confidentiality Offset : 0

Capabilities

ICV length : 16  
Data length change supported: yes  
Max. Rx SA : 16  
Max. Tx SA : 16  
Max. Rx SC : 8  
Max. Tx SC : 8  
Validate Frames : strict  
PN threshold notification support : Yes  
Ciphers supported : GCM-AES-128

GCM-AES-256

GCM-AES-XPN-128

GCM-AES-XPN-256

Transmit Secure Channels

SCI : 0CD0F8DCDC010008  
SC state : notInUse(2)

Elapsed time : 00:24:38

Start time : 7w0d  
Current AN: 0  
Previous AN: -  
Next PN: 2514  
SA State: notInUse(2)  
Confidentiality : yes  
SAK Unchanged : yes  
  
SA Create time : 1d01h

SA Start time : 7w0d

#### SC Statistics

Auth-only Pkts : 0  
Auth-only Bytes : 0  
  
Encrypt Pkts : 3156 <-- should increment with Tx traffic

Encrypt Bytes : 0

#### SA Statistics

Auth-only Pkts : 0  
  
Encrypt Pkts : 402 <-- should increment with Tx traffic

#### Port Statistics

Egress untag pkts 0  
Egress long pkts 0

#### Receive Secure Channels

SCI : A0F8490EA91F0026  
SC state : notInUse(2)  
  
Elapsed time : 00:24:38

Start time : 7w0d  
Current AN: 0  
Previous AN: -  
Next PN: 94  
RX SA Count: 0  
SA State: notInUse(2)  
SAK Unchanged : yes  
SA Create time : 1d01h  
SA Start time : 7w0d

SC Statistics

Notvalid pkts 0
Invalid pkts 0
Valid pkts 0
Valid bytes 0
Late pkts 0
Uncheck pkts 0
Delay pkts 0
UnusedSA pkts 0
NousingSA pkts 0
Decrypt bytes 0

SA Statistics

Notvalid pkts 0
Invalid pkts 0
Valid pkts 93
UnusedSA pkts 0
NousingSA pkts 0
!

Port Statistics

Ingress untag pkts 0
Ingress notag pkts 748
Ingress badtag pkts 0
Ingress unknownSCI pkts 0
Ingress noSCI pkts 0
Ingress overrun pkts 0

C9500#

sh mka sessions interface fortyGigabitEthernet 1/0/1

Summary of All Currently Active MKA Sessions on Interface FortyGigabitEthernet1/0/1...

Table with columns: Interface, Local-TxSCI, Policy-Name, Inherited, Key-Server, Port-ID, Peer-RxSCI, MACsec-Peers, Status, CKN. Row 1: Fo1/0/1, 0cd0.f8dc.dc01/0008

MKA

NO YES

```
8          a0f8.490e.a91f/0026      1          Secured01      <-- CKN number must match on both sides
```

```
0cd0.f8dc.dc01
```

```
<--
```

```
MAC of local interface
```

```
a0f8.490e.a91f
```

```
<--
```

```
MAC of remote neighbor
```

```
8
```

```
<-- indicates IIF_ID of respective local port (here IF_ID is 8 for local port fo1/0/1)
```

```
C9500#
```

```
sh platform pm interface-numbers | in iif|1/0/1
```

```
interface
```

```
iif-id
```

```
gid slot unit slun HWIDB-Ptr status status2 state snmp-if-index  
Fo1/0/1
```

```
8
```

```
1 1 1 1 0x7EFF3F442778 0x10040 0x20001B 0x4 8
```

```
C9500#
```

```
sh mka sessions interface fortyGigabitEthernet 1/0/1 detail
```

```
MKA Detailed Status for MKA Session
```

```
=====
```

```
Status: SECURED - Secured MKA Session with MACsec
```

```
Local Tx-SCI..... 0cd0.f8dc.dc01/0008
```

```
Interface MAC Address.... 0cd0.f8dc.dc01
```

```
MKA Port Identifier..... 8
```



Interface Name..... FortyGigabitEthernet1/0/1

Audit Session ID.....

CAK Name (CKN)..... 01

Member Identifier (MI)... DFDC62E026E0712F0F096392

Message Number (MN)..... 536 <-- should increment as message numbers increment

EAP Role..... NA

Key Server..... YES

MKA Cipher Suite..... AES-256-CMAC

Latest SAK Status..... Rx & Tx

Latest SAK AN..... 0

Latest SAK KI (KN)..... DFDC62E026E0712F0F09639200000001 (1)

Old SAK Status..... FIRST-SAK

Old SAK AN..... 0

Old SAK KI (KN)..... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)

SAK Retire Time..... 0s (No Old SAK to retire)

SAK Rekey Time..... 0s (SAK Rekey interval not applicable)

MKA Policy Name..... MKA

Key Server Priority..... 200

Delay Protection..... NO

Delay Protection Timer..... 0s (Not enabled)

Confidentiality Offset... 0

Algorithm Agility..... 80C201

SAK Rekey On Live Peer Loss..... NO

Send Secure Announcement.. DISABLED

SAK Cipher Suite..... 0080C20001000002 (GCM-AES-256)

MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)

MACsec Desired..... YES

# of MACsec Capable Live Peers..... 1 <-- Peers capable of MACsec

# of MACsec Capable Live Peers Responded.. 1 <-- Peers that responded to MACsec negotiation

Live Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority	RxSA Installed
ACF0BD8ECCA391A197F4DF6B	537	a0f8.490e.a91f/0026	200	YES <-- One live peer

!

Potential Peers List:

```

MI                MN    Rx-SCI (Peer)          KS          RxSA
                Priority          Installed
-----

```

Check the MKA policy and ensure that it is applied to expected interface

C9500#

sh mka policy MKA

MKA Policy defaults :

Send-Secure-Announcements: DISABLED

!

MKA Policy Summary...

!

Codes : CO - Confidentiality Offset, ICVIND - Include ICV-Indicator,

SAKR OLPL - SAK-Rekey On-Live-Peer-Loss,

DP - Delay Protect, KS Prio - Key Server Priority

Policy

Name	KS	DP	CO	SAKR	ICVIND	Cipher	Interfaces
	Prio		OLPL			Suite(s)	Applied
=====							

MKA

200 FALSE 0 FALSE TRUE

GCM-AES-256

Fo1/0/1 <-- Applied to Fo1/0/1

### Ensure that PDU counters are incrementing at Tx/Rx at both sides.

This is useful to determine the direction of issues at transport. ###

C9500#

sh mka statistics | sec PDU

MKPDU Statistics

MKPDUs Validated & Rx..... 2342 <-- should increment

"Distributed SAK"..... 0  
"Distributed CAK"..... 0  
MKPDUs Transmitted..... 4552 <-- should increment

### MKA Error Counters ###

C9500#

show mka statistics

\*\* snip\*\*\*

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

### Stap 3 tot en met stap 5

Volg dezelfde instructies als vermeld in scenario 1

**Waarschuwing: voor interoperabiliteitsdoeleinden.** Houd er rekening mee dat sommige platforms wel opvulling en sommige platforms niet doen, dus dit kan leiden tot belangrijke problemen waar de mka sessie in "Init" staat. U kunt dit verifiëren met "**mka sessies tonen**"

### Voorbeeld van opvulling

Deze use case toont een Catalyst 9500 en een Nexus 7k in NX-OS 8.2(2) maar kan ook gebeuren met Catalyst-apparaten zoals C3560CX.

(Het probleem wordt gedocumenteerd met Cisco bug-ID [CSCvs92023](#)).



- Als u de configuratie volgt die in scenario 2 wordt voorgesteld, zal MKA niet de tunnel vestigen toe te schrijven aan een zeer belangrijke wanverhouding.
- U moet de sleutel handmatig invullen met 0's aan de 9500 kant omdat dit apparaat geen opvulling.

### Catalyst 9500

```
<#root>  
  
conf t  
  key chain macsec1 macsec  
    key  
  
010000000000000000000000000000000000000000000000000000000000000000000000 --> device does not do padding automati  
  
  key-string 12345678901234567890123456789012  
end
```

### Nexus 7k

```
<#root>  
  
conf t
```

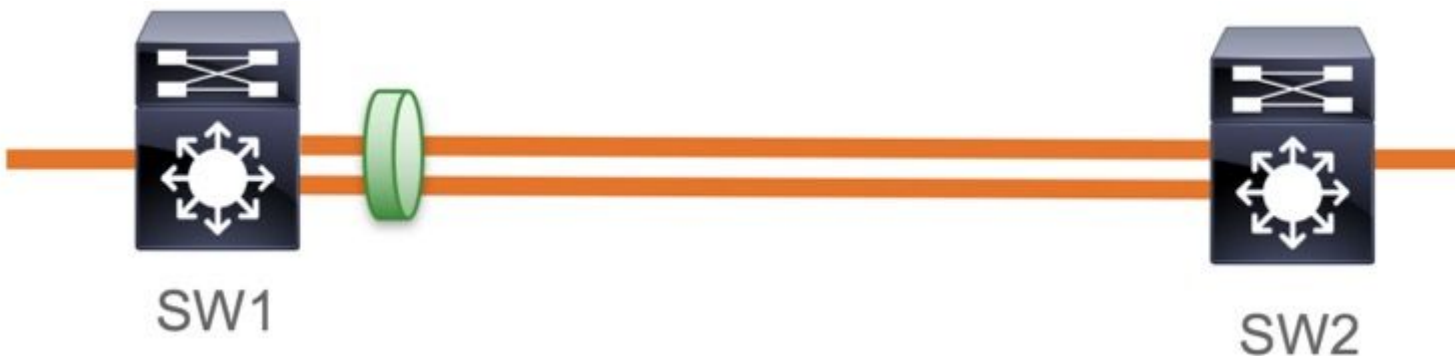
```
key chain macsec1 macsec
```

```
key 01 --> Device does automatic padding.
```

```
key-octet-string 12345678901234567890123456789012  
end
```

## Andere configuratieopties

### MACsec Switch-to-Switch Link-beveiliging met MKA op gebundelde/poortkanaals interface



- L3- en L2-poortkanalen (LACP, PAgP en Mode ON)
- Encryptietypen (AES-128 en AES-256 (AES-256 is van toepassing voor Advantage-licentie))
- Key Exchange MKA alleen PSK

Ondersteunde platforms:

- Catalyst 9200 (alleen AES-128)
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500 en Catalyst 9500H switch
- Catalyst 9600

### Configuratie Switch naar Switch EtherChannel

De sleutelketen en de MKA beleidsconfiguratie blijven hetzelfde zoals eerder getoond in MKA configuratie sectie.

```
<#root>
```

```
interface <> <-- This is the physical member link. MACsec encrypts on the individual links
```

```
macsec network-link
```

```
mka policy <policy-name>  
mka pre-shared-key key-chain <key-chain name>  
macsec replay-protection window-size frame number
```

channel-group

```
mode active <-- Adding physical member to the port-channel
```

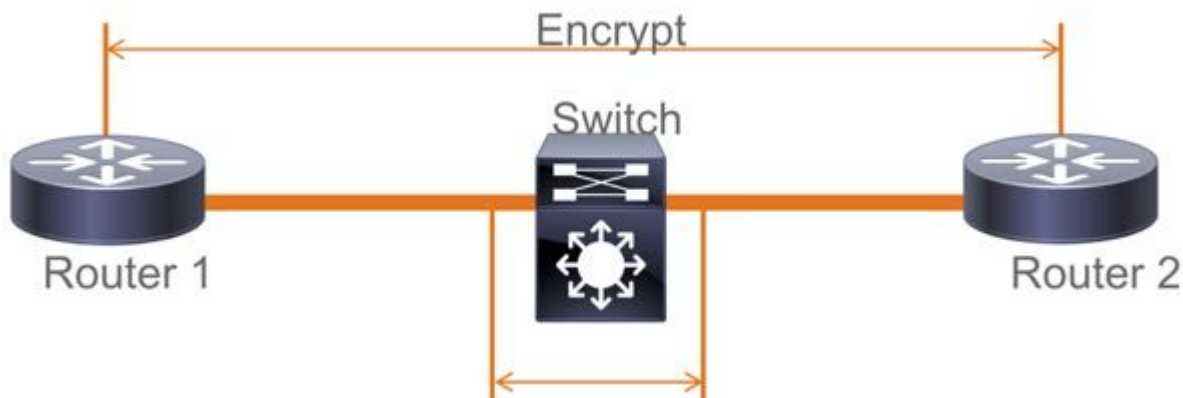
## MACsec Switch-to-Switch Link Security via L2 tussenliggende switches, PSK-modus

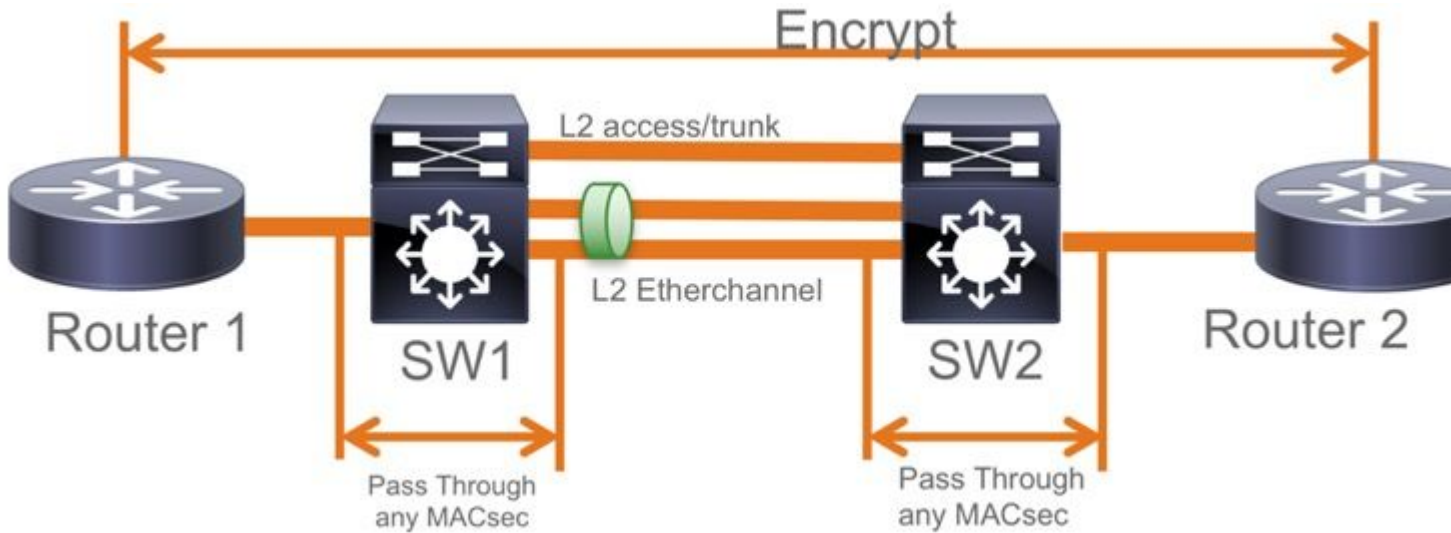
Deze paragraaf behandelt een aantal van die ondersteunde WAN MACsec-scenario's waarbij Cat9K versleutelde pakketten transparant moet doorgeven.

Er zijn gevallen waarin routers niet direct zijn aangesloten maar ze hebben L2 tussenpakketten, en de L2 switches moeten de versleutelde switches omzeilen zonder enige bewerking van de versleuteling.

### Catalyst 9000 switches voorwaartse transparante pakketten met Clear Tag vanaf 16.10(1)

- Pass through wordt ondersteund voor MKA/SAP
- Ondersteund op L2 access, trunk of Ethernet kanalen
- Standaard ondersteund (geen configuratie-CLI's om in/uit te schakelen)
- **Zorg ervoor dat routers EAPOL-frames zonder standaard (0x88E) verzenden**

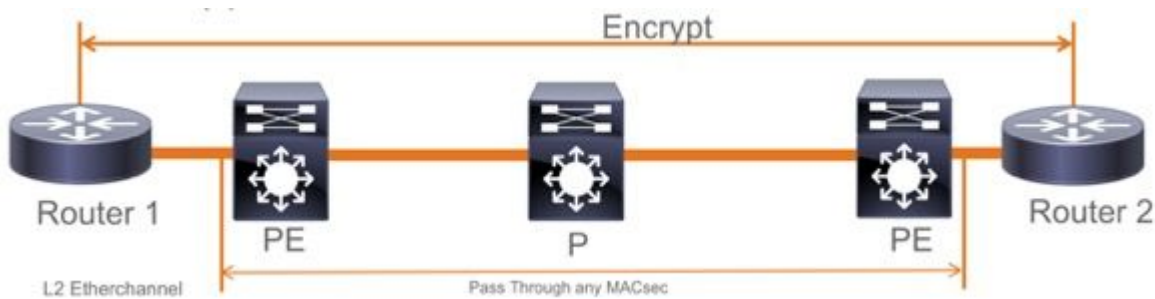




### EoMPLS/VPLS-topologie

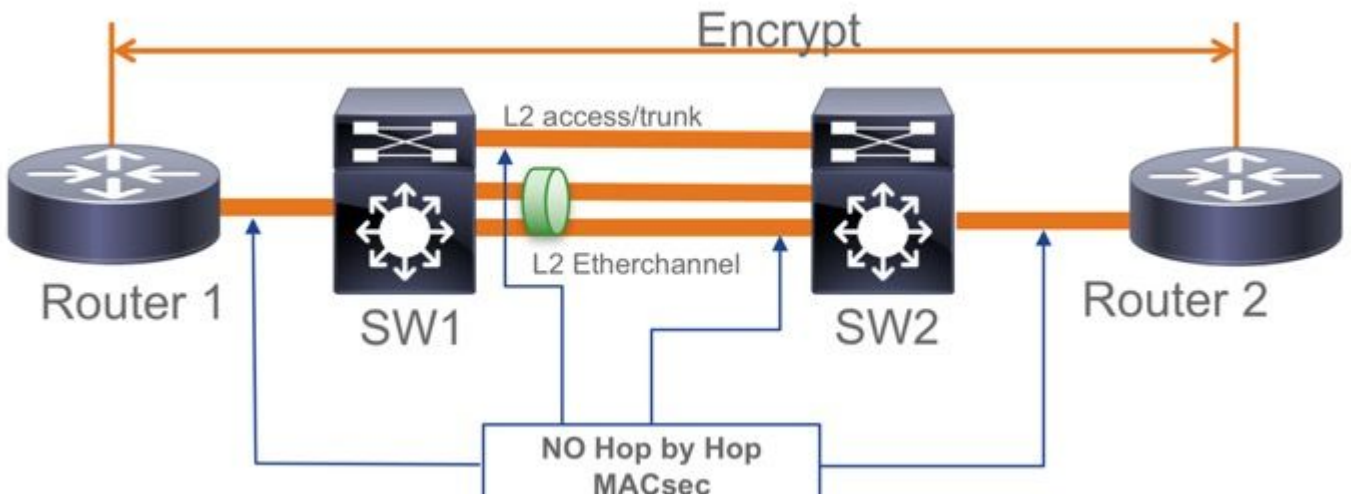
Ondersteunde platforms Cat 9300/9400,9500/9500H als "PE" of "P" apparaten

- VPLS
- EoMPLS
- Standaard ondersteund (geen configuratie-CLI's om in/uit te schakelen)
- Start 16.10(1)

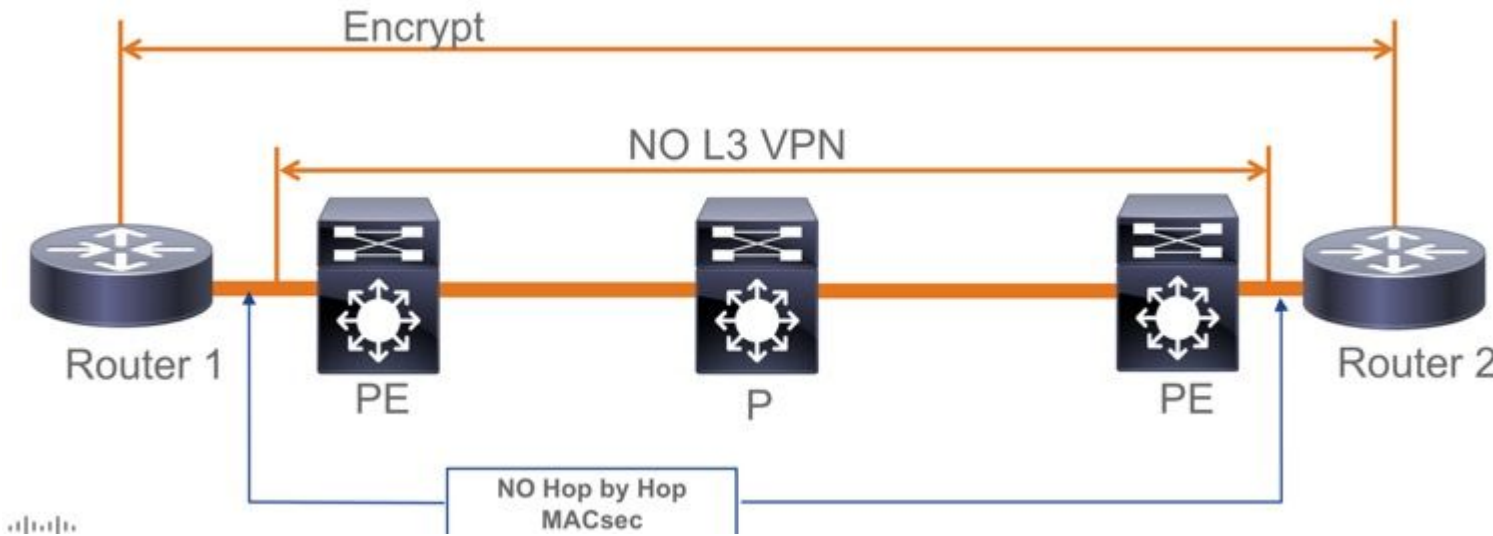


### Beperkingen

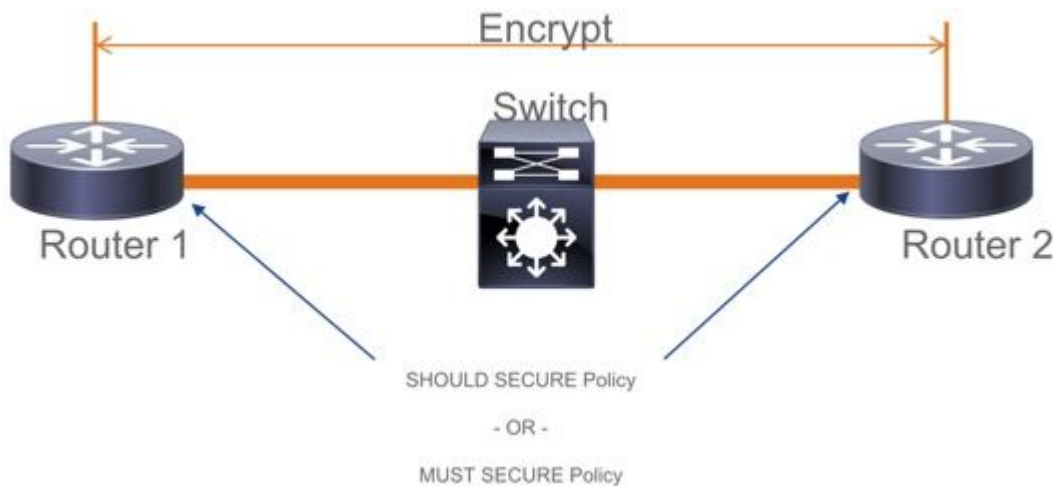
Dubbele codering wordt niet ondersteund. End-to-end MACsec met Clear tag vereist dat de Hop by Hop switches niet op de L2 direct verbonden links inschakelen



- ClearTag + EoMPLS met tussenliggende Layer 2-switches, MACsec kan niet inschakelen op CE-PE-link
- ClearTag + L3VPN met tussenliggende switches niet ondersteund



- Er is geen ondersteuning voor "Moeten beveiligen" in PSK-modus, "Moeten beveiligen" is de standaardmodus
- Moet Beveiligd beleid niet alleen EAPoL versleutelen om de MACsec-instellingen te onderhandelen



## Operationele informatie over MACsec

### Bedieningssequentie

1. Wanneer de link en beide eindapparaten omhoog komen, ruilen ze MKA frames (**ethertype = 0x888E**, hetzelfde als EAPOL met pakkettype als MKA). Het is een multipoint-to-multipoint onderhandelingsprotocol. De CAK-sleutelwaarde (normaal statisch preshared), toetsnaam (CKN) moet overeenkomen en ICV moet geldig zijn voor peers die moeten worden ontdekt en geaccepteerd.
2. Het apparaat met de laagste Key Server-prioriteit (standaard = 0) wordt geselecteerd als de Key Server. De Key-server genereert de SAK en verspreidt via MKA-berichten. In geval van tijd hoogste waarde van SCI (Secure Channel Identifier) wint.
3. Vervolgens worden alle door MacSec beveiligde frames versleuteld met de SAK (symmetrische cryptografie). Er zijn aparte TX en RX beveiligde kanalen gemaakt. Maar dezelfde Key SAK wordt gebruikt voor zowel versleutelen als ontsleutelen.
4. Wanneer een nieuw apparaat wordt gedetecteerd in een multi-access LAN (via EAPOL-MKA-



berichten) genereert de sleutelserver een nieuwe sleutel die door alle apparaten moet worden gebruikt. De nieuwe sleutel wordt in gebruik genomen nadat deze door alle apparaten is bevestigd (zie paragraaf 9.17.2 van de IEEE Std 802.1X-2010).



## MACsec-pakketten

### Bedieningskader (EAPOL-MKA)

- EAPOL-bestemming MAC = 01:80:C2:00:00:03 voor multicast van de pakketten naar meerdere bestemmingen
- EAPOL-ethertype = 0x888E

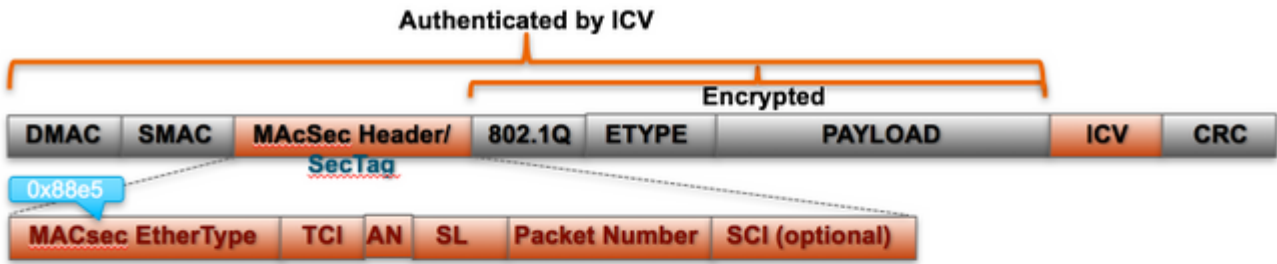
### L2 payload in het formaat van het controleframe

Protocol Version		
Packet Type = EAPOL-MKA		
Packet Body Length		Size
Packet Body (MKPDU)	Basic Parameter Set	Multiple of 4 octets
	Parameter Set	Multiple of 4 octets
	Parameter Set	Multiple of 4 octets
	ICV	16 octets

### Gegevenskader

MACSec voegt twee extra tags toe aan gegevenskaders met een maximale overhead van **32 bytes** (min. 16 bytes).

- **SecTag** = 8 tot 16 bytes (8 bytes SCI is optioneel)
- **ICV** = 8 tot 16 bytes op basis van het algoritme (AES128/256)



**MACsec Tag Format**

Field	Size	Description
Ethertype	16 bit	MAC length/type value for MACsec packet EtherType = 88-E5
TCI	6 bit	Tag control info contains: Version, ES, SC, SCB, E, C (indicates how frame is protected)
AN	2 bit	Association number
SL	8 bit	Short Length Indicates MSDU length of 1-48 octets 0 indicates MSDU length > 48 octets
PN	32 bit	Packet sequence number
SCI	64 bit	Secure channel identified (optional)

## SAP-onderhandeling

# SAP Negotiation



### Pair-wise Master Key (PMK)

(Manually configured or derived through 802.1X authentication)



PMK is never sent on the link



**Role determination:** Lowest MAC = Authenticator (Manual Mode), RADIUS server tells who is who (802.1X Mode)



Authenticator and Supplicant derive keys and exchange with each other

$PMKID(16) = HMAC-SHA1-128(PMK, "PMK Name" || AA || SA)$

*AA: Authenticator Address, SA: Supplicant Address*

$PTK \leftarrow PRF-X(PMK, "Pairwise key expansion", \text{Min}(AA, SA) || \text{Max}(AA, SA) || \text{Min}(ANonce, SNonce) || \text{Max}(ANonce, SNonce))$

*ANonce & SNonce = Random values gen by Authenticator & Supplicant respectively*

Pairwise Transient Key PTK

Key Confirmation Key (KCK)

Key Encryption Key (KEK)

Temporal Key (TK)

Message Integrity check (16) Encryption Alg (16)

Data Encryption

AUTHENTICATOR  
BLDG-1-AGG



EAPoL-

EAPoL-

EAPoL-Key (

EAPoL-Key (S

EAPoL-Key (

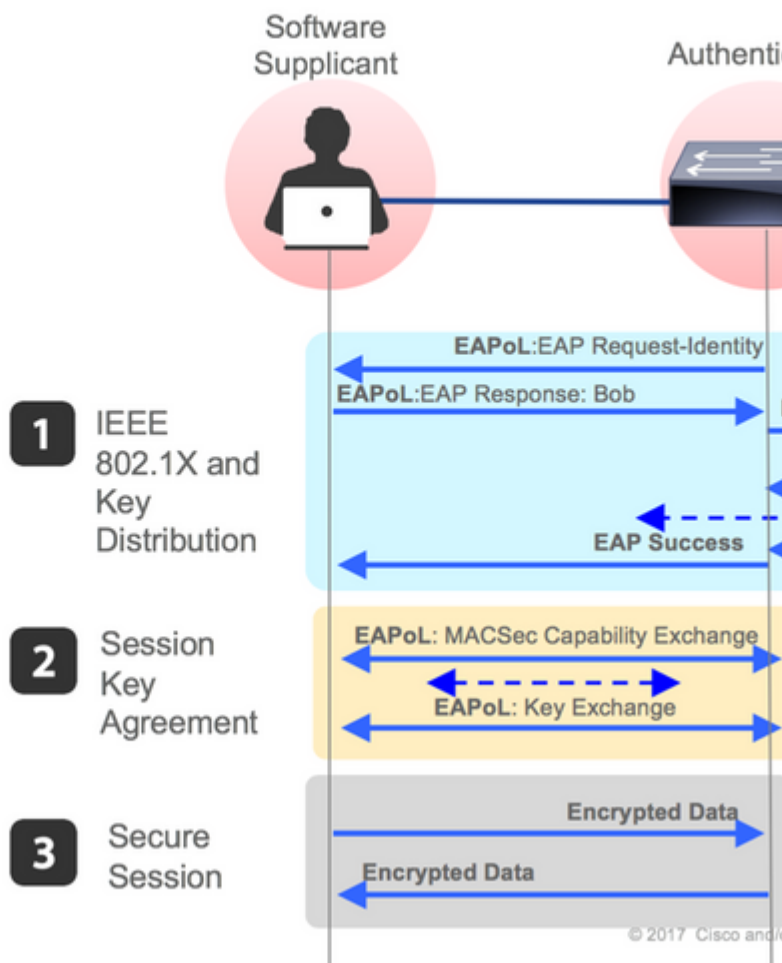
EAPoL-

# MACsec Key Derivation Schemes

**Session Key Agreement Protocols**

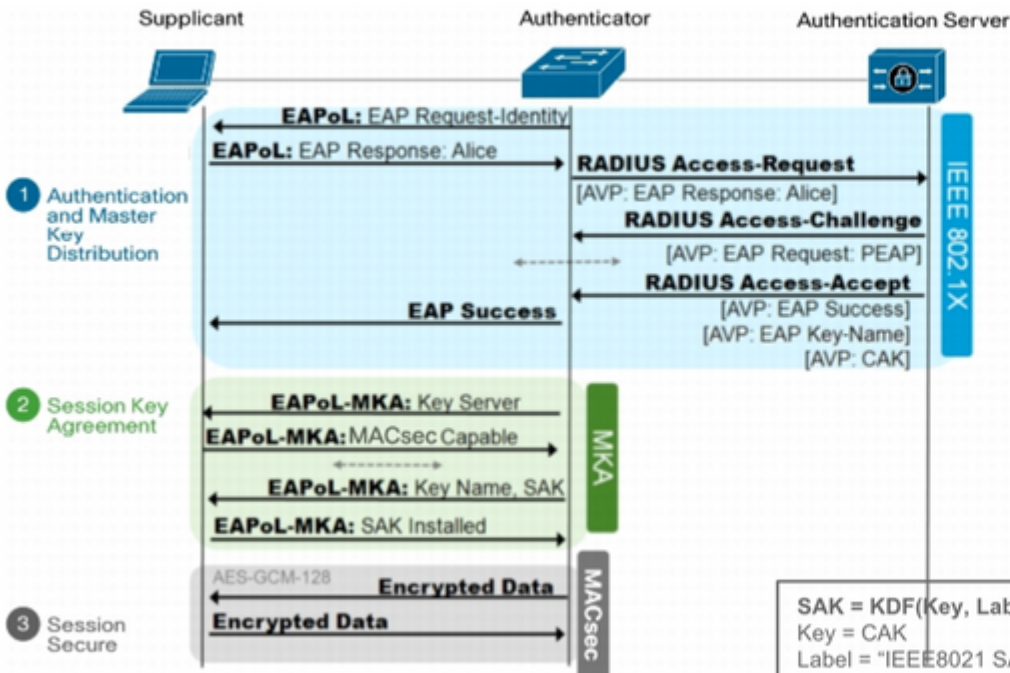
- SAP** **Security Association Protocol** is Cisco proprietary protocol for MACSec Key negotiation.  
Used only for Switch-to-Switch encryptions.
- MKA** **MKA (MACsec Key Agreement)** is defined in IEEE 802.1X-2010.  
Used today for Switch-to-Host encryptions. Router MACsec uses MKA

CISCO



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

# MKA Exchange



A pairwise CAK (Connectivity Association Key) is derived from the following parameters:  
**CAK = KDF(Key, Label, mac1 | mac2)**  
 Key = MSK[0-15] for a 128 bit CAK, MSK[16-31] for a 256 bit CAK  
 Label = "IEEE8021 EAP CAK"  
 mac1 = the lesser of the two source MAC addresses  
 mac2 = the greater of the two source MAC addresses  
 CAKLength = two octets representing an integer value (128 for a 128 bit CAK, 256 for a 256 bit CAK) with the most significant octet first.

The KEK (Key Encryption Key) is derived from the following parameters:  
**KEK = KDF(Key, Label, Keyid, KEKLength)**

Key = CAK  
 Label = "IEEE8021 KEK"  
 Keyid = the first 16 octets of the CKN, with the most significant octet first  
 KEKLength = two octets representing an integer value (128 for a 128 bit KEK, 256 for a 256 bit KEK) with the most significant octet first.

The ICK (ICV Key) is derived from the following parameters:

**ICK = KDF(Key, Label, Keyid, ICKLength)**

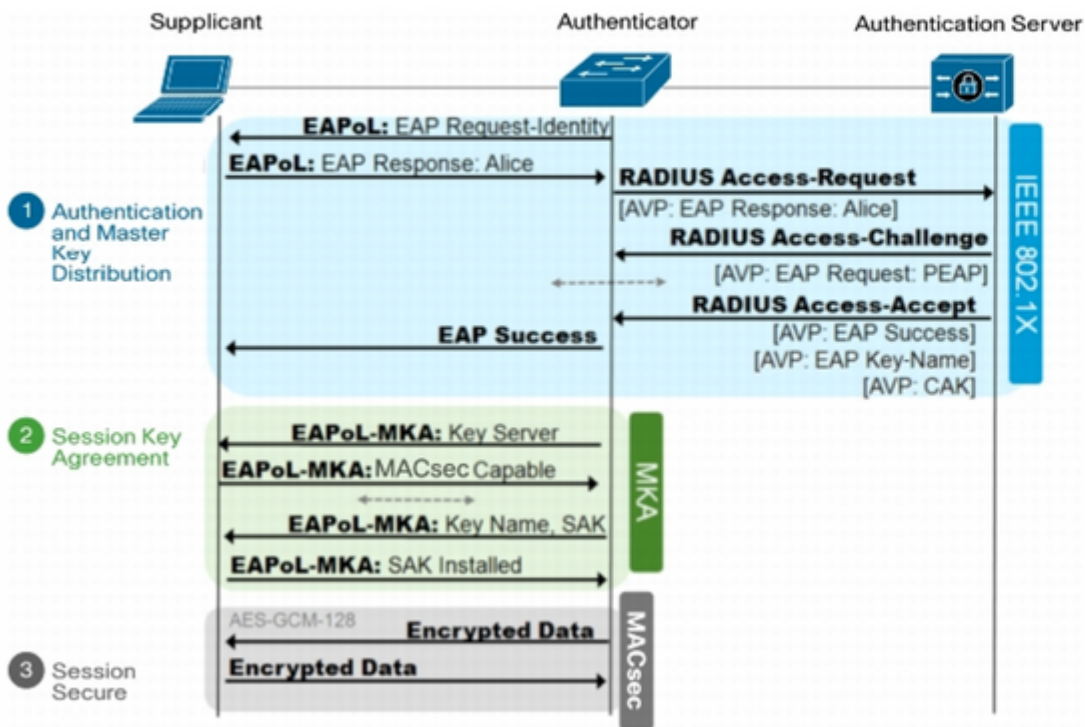
Key = CAK  
 Label = "IEEE8021 ICK"  
 Keyid = the first 16 octets of the CKN, with the most significant octet first  
 ICKLength = two octets representing an integer value (128 for a 128 bit ICK, 256 for a 256 bit ICK) with the most significant octet first.

**SAK = KDF(Key, Label, KS-nonce | MI-value list | KN, SAKLength)**  
 Key = CAK  
 Label = "IEEE8021 SAK"  
 KS-nonce = a nonce of the same size as the required SAK, obtained from the Key Server  
 MI-value list = a concatenation of MI values (in no particular order)  
 KN = four octets, the Key Number assigned by the Key Server as part of the RADIUS Access-Accept  
 SAKLength = two octets representing an integer value (128 for a 128 bit SAK, 256 for a 256 bit SAK) with the most significant octet first.

$$ICV = AES-CMAC(ICK, M, 128)$$

$$M = DA + SA + (MSDU - ICV)$$

# MKA Exchange



MKA  
\* 802  
\* Pre

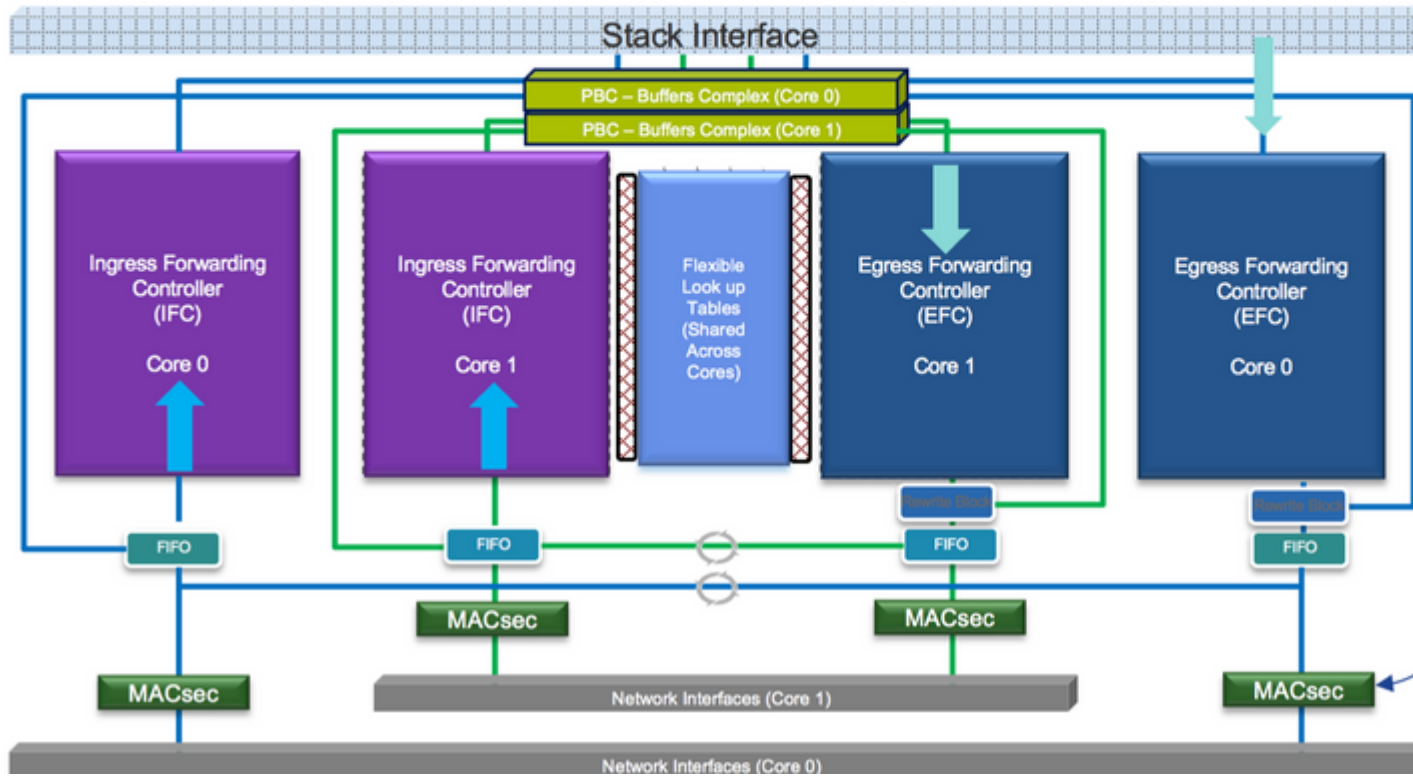


MKA  
\* Rec  
\* ISE  
\* 802

MACsec op platform

# Where is MACsec performed in Hardware?

Applicable for UADP 2.0/3.0/Mini ASIC



## Productcompatibilitestabel

## LAN MACsec Support per Platform

	MACsec	Cat 9200		Cat 9300		Cat 9400		Cat 9500
		SW	License	SW	License	SW	License	SW
<b>Switch to Switch</b>	128 Bits SAP	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +
	128 Bits MKA	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +
	256 Bits MKA	Not Supported		16.6.1 +	NA	16.10.1 +	NA	16.6.1 +
	ClearTag Pass Through	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +
<b>Host to Switch</b>	128 Bits MKA	16.10.1 +	NE	16.8.1 +	NE	16.9.1 +	NE	16.8.1 +
	256 Bits MKA	Not Supported		16.9.1 +	NA	16.10.1 +	NA	16.9.1 +

NE – Network Essentials. NA – Network Advantage.

**C9300 Stackwise 480 / C9500 SWV High Availability is not supported for MACsec**

**C9400 Sup 1XL-Y does not Support MACsec on any Supervisor ports**

**C9400 Sup 1 and 1XL support MACsec for only for interfaces with speed 10/40 Gbps**

# LAN MACsec Performance Data

	MACsec	Cat 9200	Cat 9300	Cat 9400	Cat 9500
Switch to Switch	128 Bits SAP	Line Rate	Line Rate	Line Rate	Line Rate
	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	Line Rate	Line Rate
Host to Switch	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	Line Rate	Line Rate

**C9400 Sup 1XL-Y does not Support MACsec on any Supervisor ports**  
**C9400 Sup 1 and 1XL support MACsec for only for interfaces with speed 10/40**

NE – Network Essentials. NA – Network Advantage.  
Line rate is calculated with the additional MACsec header overhead

## Gerelateerde informatie

[Security Configuration Guide, Cisco IOS XE Gibraltar 16.12.x \(Catalyst 9300 Switches\)](#)



## Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.