

Catalyst 9000上的MACsec故障排除

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[MACsec的優點](#)

[MACsec和MTU](#)

[使用MACsec的位置](#)

[技術](#)

[案例1：在預共用金鑰\(PSK\)模式下使用SAP的MACsec交換機到交換機鏈路安全](#)

[拓撲](#)

[場景2：在預共用金鑰\(PSK\)模式下使用MKA的MACsec交換機到交換機鏈路安全](#)

[拓撲](#)

[填充問題示例](#)

[其他組態選項](#)

[在捆綁式/埠通道介面上使用MKA的MACsec交換機到交換機鏈路安全](#)

[第2層中間交換機之間的MACsec交換機到交換機鏈路安全 - PSK模式](#)

[約束](#)

[MACsec操作資訊](#)

[操作順序](#)

[MACsec資料包](#)

[SAP協商](#)

[金鑰交換](#)

[平台上的MACsec](#)

[產品相容性矩陣](#)

[相關資訊](#)

簡介

本檔案將說明MACsec功能、其使用案例，以及如何對Catalyst 9000交換器上的功能進行疑難排解。


必要條件

需求

本文件沒有特定需求。

採用元件

- C9300
- C9400
- C9500
- C9600

 注意：有關用於在其他Cisco平台上啟用這些功能的命令，請參閱相應的配置指南。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

本文的作用域為兩台交換器/路由器之間LAN上的媒體存取安全控制(MACsec)。

明文資料通訊容易受到安全威脅。安全漏洞可能發生在OSI模型的任何層。第2層的一些常見漏洞是監聽、資料包竊聽、篡改、注入、MAC地址欺騙、ARP欺騙、針對DHCP伺服器的拒絕服務(DoS)攻擊以及VLAN跳躍。

MACsec是IEEE 802.1AE標準中描述的一種L2加密技術。MACsec可以保護物理介質上的資料，並且使資料不可能在更高層受到危害。因此，MACsec加密比任何其他高層加密方法（如IPsec和SSL）的優先順序更高。

MACsec的優點

面向客戶端的模式：MACsec用於這樣的設定，即相互對等的兩台交換機在交換金鑰之前可以交替作為金鑰伺服器或金鑰客戶端。金鑰伺服器在兩個對等體之間生成並維護CAK。

資料完整性檢查:MACsec使用MKA為到達埠的幀生成完整性檢查值(ICV)。如果生成的ICV與幀中的ICV相同，則接受該幀；否則丟棄該幀。


資料加密：MACsec在交換機的介面上提供埠級加密。這表示從已設定的連線埠傳送的訊框已加密，並在連線埠上接收的訊框已解密。MACsec還提供一種機制，在該機制中您可以配置是僅加密幀還是所有加密幀

介面上接受幀（加密的和純的）。

重播保護：當幀通過網路傳輸時，可能會出現幀從有序序列中脫離的情況。MACsec提供一個可配置的視窗，該視窗接受指定數量的亂序幀。

MACsec和MTU

MACsec報頭增加了多達32位元組的報頭開銷。請考慮路徑中交換器上較大的系統/介面最大傳輸單元(MTU)，以解決MACsec標頭增加的額外額外負荷。如果MTU太低，對於需要使用更高MTU的應用程式，您可能會看到意外的資料包丟失/延遲。

 註：如果存在與MACsec相關的問題，請確保根據相容性表支援兩端的千兆位介面轉換器 (GBIC)。

使用MACsec的位置

園區使用案例

- 主機到交換機
- 在站點或建築之間
- 多租戶中的樓層之間

資料中心使用案例

- 資料中心互連
- 伺服器到交換機

WAN使用案例

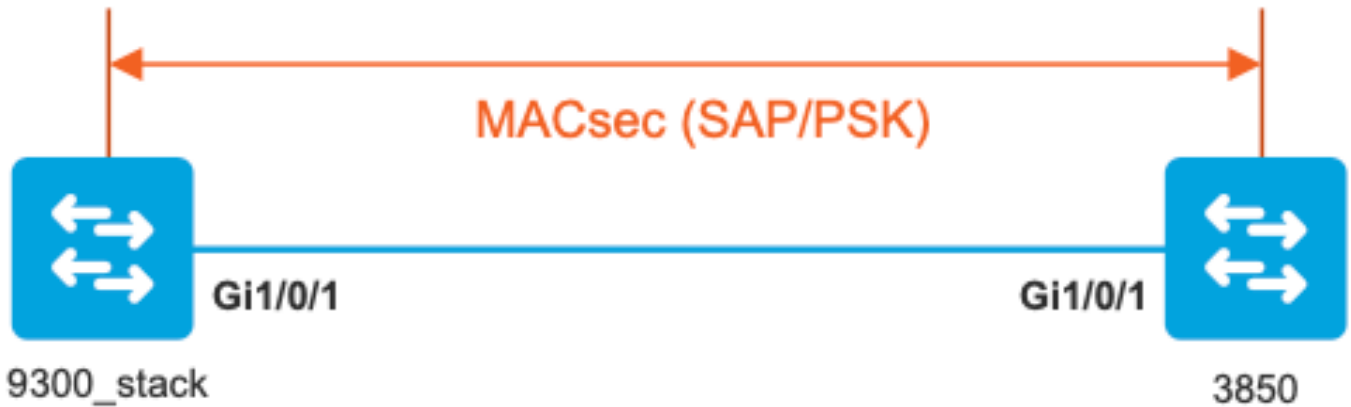
- 資料中心互連
- 園區互連
- 中心輻射型

技術

MKA	MACsec金鑰協定	在IEEE 802.1X REV-2010中定義為用於發現MACsec對等體和協商金鑰的關鍵協定協定
CAK	連線關聯金鑰	用於生成用於MACsec的所有其他金鑰的長時間主金鑰。LAN實現從MSK (在EAP交換期間生成) 派生
PMK	成對主鍵	用於派生用於加密流量的會話金鑰的元件之一。手動配置或從802.1X派生
CKN	CAK金鑰名稱	用於配置金鑰值或CAK。只允許偶數個十六進位制字元，最多64個字元。
SAK	安全關聯金鑰	由從CAK選擇的金鑰伺服器派生，是路由器/終端裝置用於加密給定會話流量的金鑰。
ICV	完整性檢查值鍵	源自CAK，並在每個資料/控制幀中標籤，以證明該幀來自授權對等體。8-16位元組，取決於密碼套件
KEK	金鑰加密金鑰	源自CAK (預共用金鑰)，用於保護MACsec金鑰
SCI	安全通道識別符號	每個虛擬埠接收唯一的安全通道識別符號(SCI)，該識別符號基於連線了16位埠ID的物理介面的MAC地址

案例1：在預共用金鑰(PSK)模式下使用SAP的MACsec交換機到交換機鏈路安全

拓撲



步驟 1. 驗證鏈路兩端的配置。

```
<#root>
```

```
9300_stack#
```

```
show run interface gig 1/0/1
```

```
interface GigabitEthernet1/0/1
description MACsec_manual_3850-2-gi1/0/1
switchport access vlan 10
switchport mode trunk
```

```
cts manual
```

```
no propagate sgt
```

```
sap pmk
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
mode-list gcm-encrypt <-- use full packet encrypt mode
```

```
3850#
```

```
show run interface gig1/0/1
```

```
interface GigabitEthernet1/0/1
description 9300-1gi1/0/1 MACsec manual
switchport access vlan 10
```

```
switchport mode trunk
```

```
cts manual
```

```
no propagate sgt
```

```
sap pmk
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

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

```
<--
```

Use the sap command to manually specify the Pairwise Primary Key (PMK) and the Security Association Prot

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 primary key (sap pmk):

SAP is not configured- no protection.
sap mode-list gcm-encrypt gmac no-encap-protection desirable but not mandatory.
sap mode-list gcm-encrypt gmac-confidentiality preferred and integrity required.
The protection is selected by the supplicant according to supplicant preference.
sap mode-list gmac -integrity only.
sap mode-list gcm-encrypt-confidentiality required.
sap mode-list gmac gcm-encrypt-integrity required and preferred, confidentiality optional.

步驟 2. 驗證MACsec狀態，以及引數/計數器是否正確。

```
<#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-XPN-128

GCM-AES-XPN-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

步驟 3. 鏈路啟動時檢查軟體調試。

<#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: 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
AA

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
AA

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: getting 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: getting 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: getting 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: getting 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: getting 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

```

步驟 4. 鏈路啟動時檢視平台級跟蹤。

```
<#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 MACsec

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 c

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

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_idMA

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}: [sec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4d01

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 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 create R

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 mac

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): txSC setting x

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 cal
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 T
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_
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending mac
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_fr
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering clear_
2019/05/08 01:08:50.527 {fed_F0-0}{1}: [pm_xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): XCVR POST:XCVR
speed_auto Oper Speed:speed_gbps1 Autoneg Mode:Unknown autonegmode type
2019/05/08 01:08:50.525 {fed_F0-0}{1}: [xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): ntfy_lnk_status:
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

```

步驟 5. 驗證硬體中MACsec介面的狀態。

```
<#root>
```

```
9300_stack#
```

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

```

interface iif-id gid slot unit slun HWIDB-Ptr status status2 state snmp-if-index
-----
Gi1/0/1 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-f
priv_ri/priv_si Handle: (nil)Hardware Indices/Handles: index1:0x0 mtu_index/13u_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
```

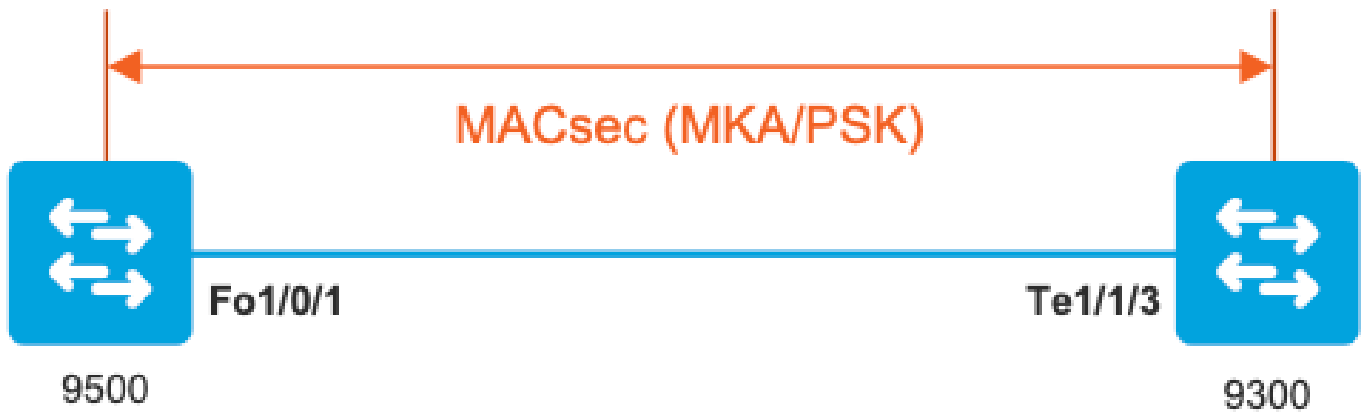
```
LEAD_PORT_EGRESS_MAC_sec_SUB_PORT value 0 Pass
```

```
LEAD_PORT_EGRESS_MACsec_ENCRYPTED value 0 Pass
```

snip

案例2：在預共用金鑰(PSK)模式下使用MKA的MACsec交換機到交換機鏈路安全

拓撲



步驟 1. 驗證鏈路兩端的配置。

```
<#root>
```

```
C9500#
```

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

```
key chain KEY MACsec
```

```
key 01
```

```
cryptographic-algorithm aes-256-cmac
```

```
key-string 7 101C0B1A0343475954532E2E767B3233214105150555030A0004500B514B175F5B05515153005E0E5E505C52
```

```
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
```

步驟2.驗證MACsec是否已啟用以及所有引數/計數器是否正確。

```
<#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-XPB-128

GCM-AES-XPB-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 <-- can increment with Tx traffic

Encrypt Bytes : 0

SA Statistics

Auth-only Pkts : 0

Encrypt Pkts : 402 <-- can 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...

Interface Local-TxSCI

Policy-Name

Inherited	Key-Server			
Port-ID	Peer-RxSCI	MACsec-Peers	Status	CKN

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 <-- can 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 Priority	RxSA 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

KS DP CO SAKR ICVIND Cipher Interfaces

Name

Prio OLPL Suite(s) Applied

MKA

200 FALSE 0 FALSE TRUE

GCM-AES-256

F01/0/1 <-- Applied to F01/0/1

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 <-- can increment

"Distributed SAK"..... 0

"Distributed CAK"..... 0

MKPDUs Transmitted..... 4552 <-- can 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

步驟3至步驟5

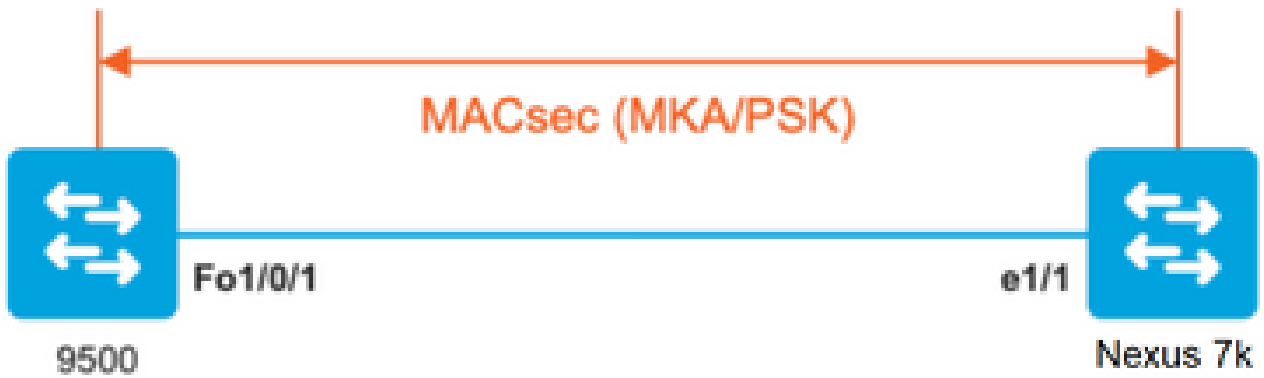
使用場景1中提到的相同說明。

 **警告：**出於互操作性的目的，請注意某些平台有填充功能，而某些平台沒有。這會導致mka會話保持初始化狀態的關鍵問題。您可以使用show mka sessions指令驗證這點。

填充問題示例

此使用案例顯示NX-OS 8.2(2)中的Catalyst 9500和Nexus 7k，但也可能與C3560CX等Catalyst裝置一起發生。

(思科錯誤ID [CSCvs92023](#)會記錄問題)。



- 如果使用場景2中顯示的配置，由於金鑰不匹配，MKA無法建立隧道。
- 由於此裝置不進行填充，您必須手動在9500端使用0完成金鑰。

Catalyst 9500

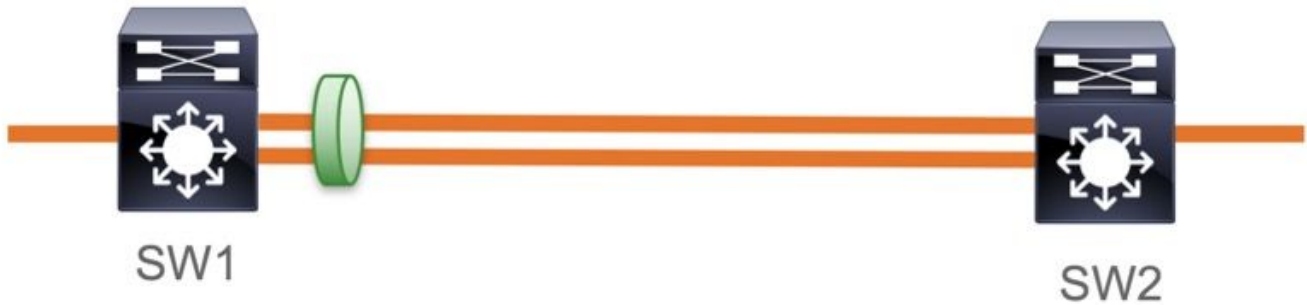
```
<#root>
conf t
  key chain MACsec1 MACsec
    key
0100000000000000000000000000000000000000000000000000000000000000 --> device does not do padding automati
    key-string 12345678901234567890123456789012
  end
```

Nexus 7k

```
<#root>
conf t
  key chain MACsec1 MACsec
key 01 --> Device does automatic padding.
    key-octet-string 12345678901234567890123456789012
  end
```

其他組態選項

在捆綁式/埠通道介面上使用MKA的MACsec交換機到交換機鏈路安全



- L3和L2埠通道 (LACP、PAgP和模式開啟)
- 加密型別 (AES-128和AES-256,AES-256適用於Advantage許可證)
- 僅限金鑰交換MKA PSK

支援的平台：

- Catalyst 9200 (僅限AES-128)
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500和Catalyst 9500H
- Catalyst 9600

交換機到交換機EtherChannel配置示例

金鑰鏈和MKA策略配置保持不變，如前面的MKA配置部分所示。

```
<#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
```

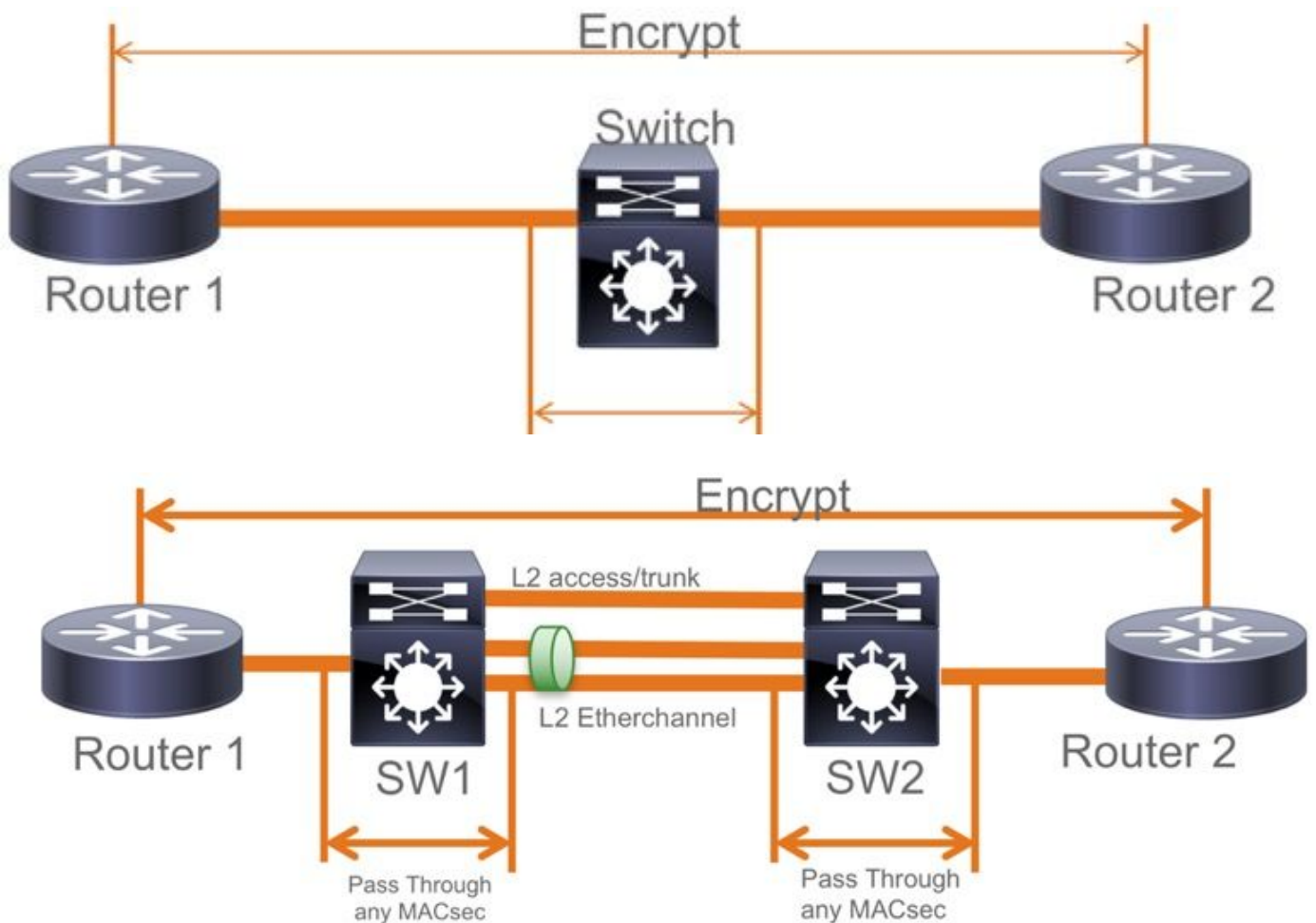
第2層中間交換機之間的MACsec交換機到交換機鏈路安全，PSK模式

本節介紹一些受支援的WAN MACsec場景，在這些場景中，Cat9K需要透明地傳遞加密資料包。

某些情況下，路由器沒有直接連線，但有L2中間交換機，並且L2交換機可以繞過加密的資料包，而無需進行任何加密處理。

Catalyst 9000交換器從16.10(1)開始透過Clear Tag轉送封包

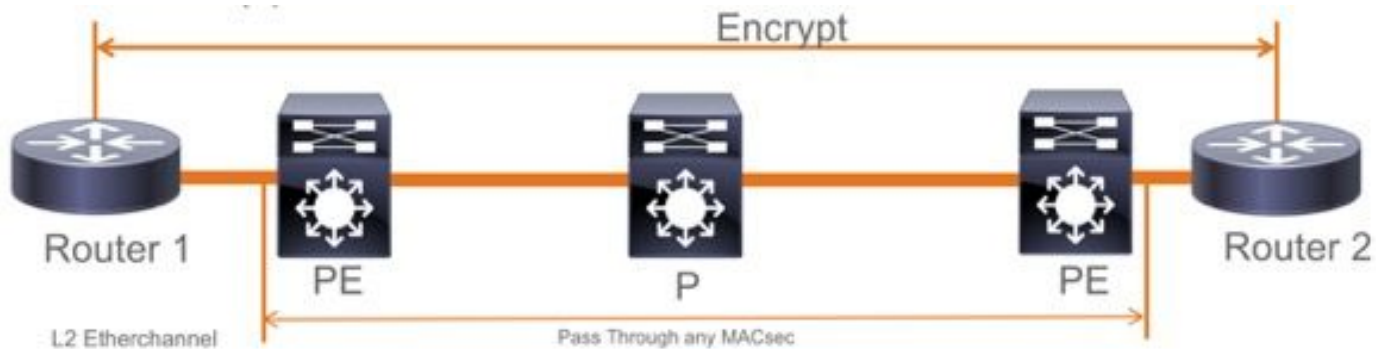
- MKA/SAP支援直通
- 在L2訪問、中繼或EtherChannel上受支援
- 預設支援 (沒有要啟用/禁用的配置CLI)
- 確保路由器傳送帶有非預設(0x888E)ether-type的EAPOL幀



EoMPLS/VPLS拓撲

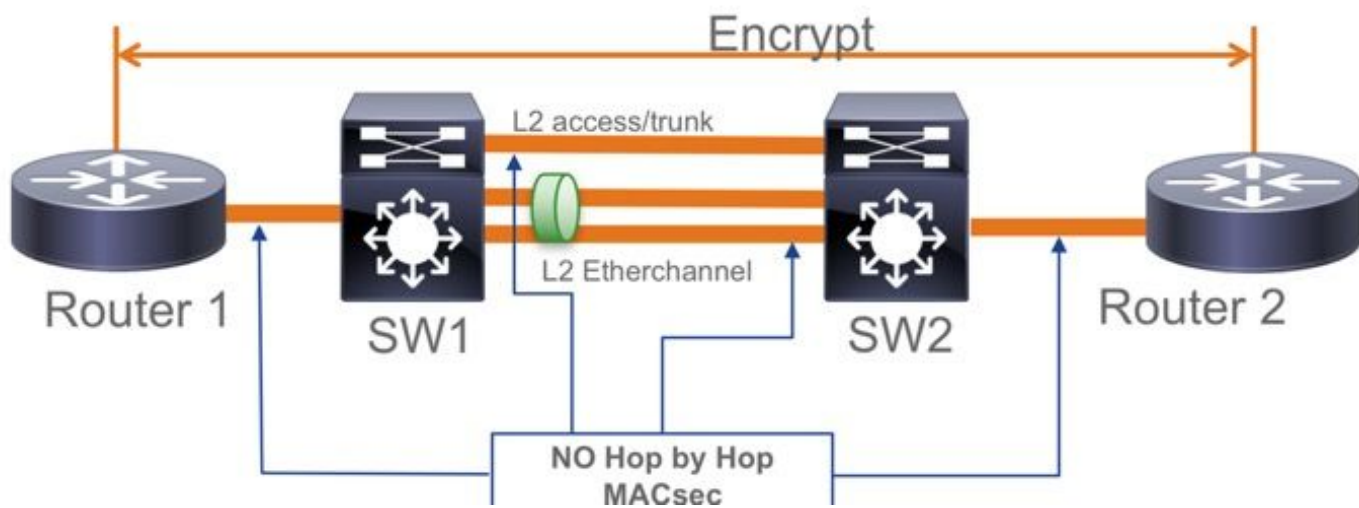
支援的平台Cat 9300/9400、9500/9500H，作為PE或P裝置

- VPLS
- EoMPLS
- 預設支援 (沒有要啟用/禁用的配置CLI)
- 啟動16.10(1)

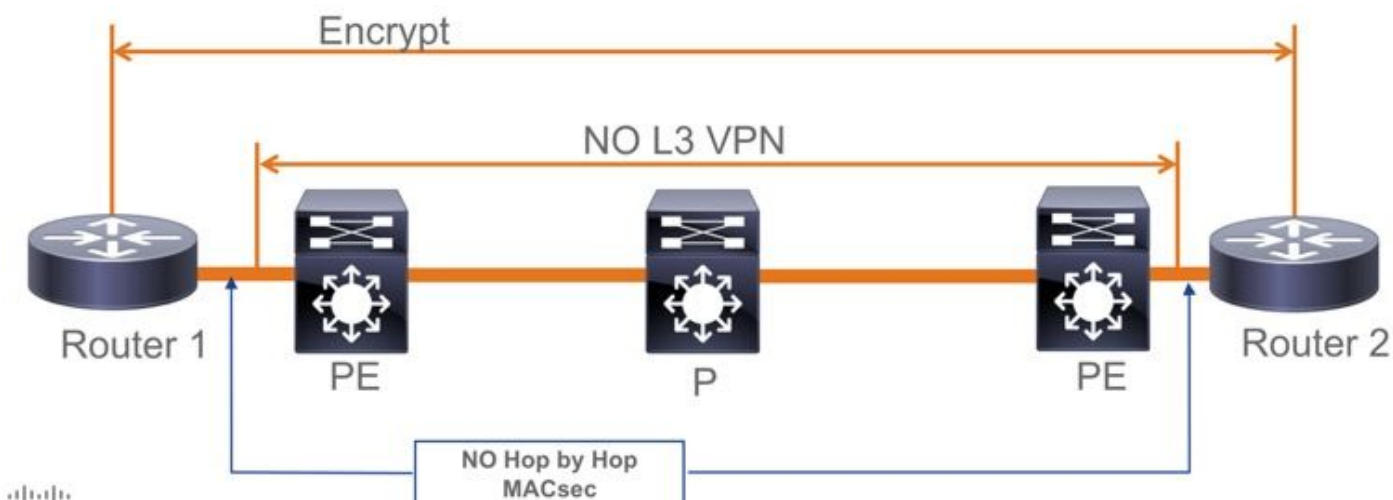


約束

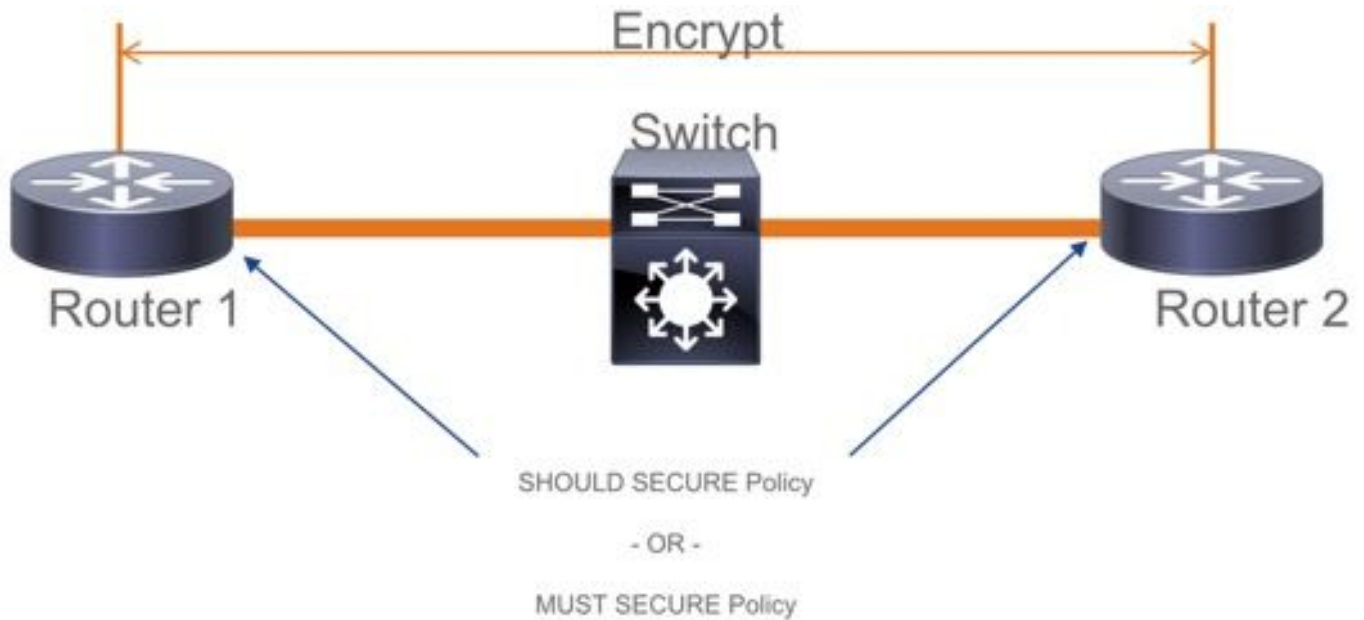
不支援雙重加密。帶Clear標籤的端到端MACsec要求不在第2層直連鏈路上啟用逐跳交換機。



- ClearTag + EoMPLS (僅使用中間第2層交換機) ， MACsec無法在CE-PE鏈路上啟用
- 不支援帶有中間交換機的ClearTag + L3VPN



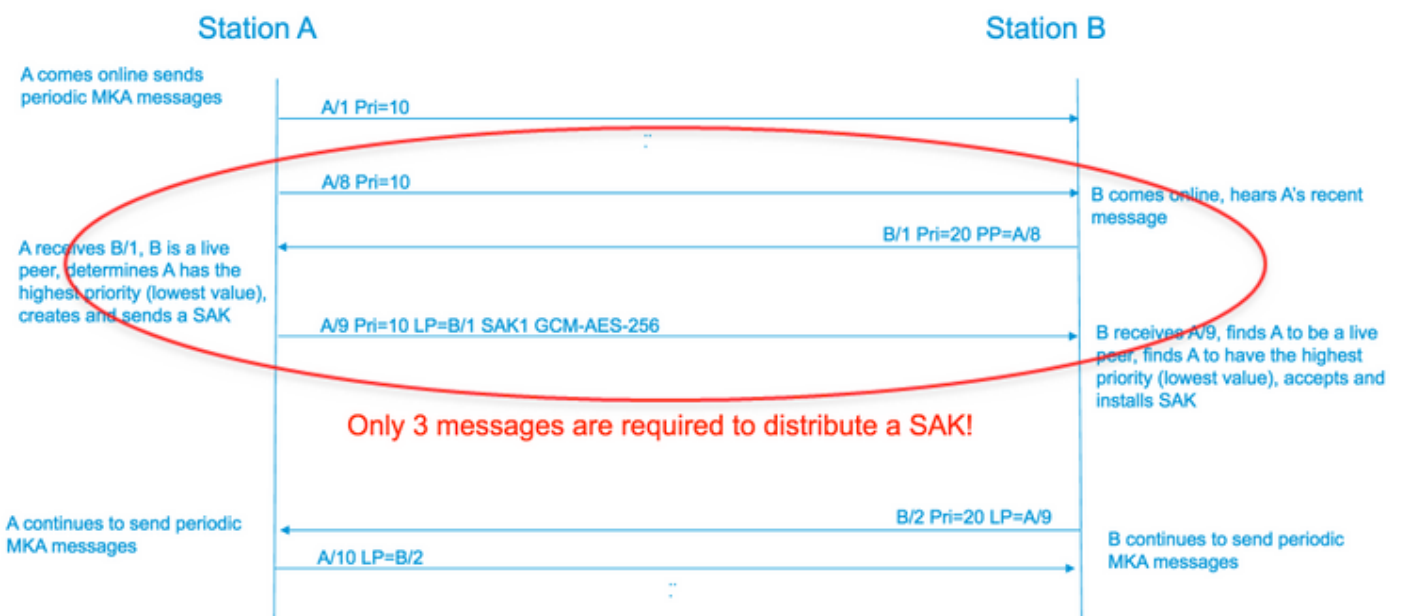
- 在PSK模式下不支援Should Secure。 Must Secure是預設模式。
- Must Secure policy不會只加密EAPoL以協商MACsec設定。



MACsec操作資訊

操作順序

1. 當鏈路和兩個終端裝置啟動時，它們交換MKA幀（ethertype = 0x888E，與資料包型別為MKA的EAPOL相同）。是一種多點到多點協商協定。CAK金鑰值（通常為靜態預共用）、金鑰名稱(CKN)必須匹配，並且ICV必須有效才能發現和接受對等體。
2. 金鑰伺服器優先順序最低的裝置（預設值= 0）被選為金鑰伺服器。金鑰伺服器生成SAK並通過MKA消息分發。如果安全通道識別符號(SCI)的值最高，則為wins。
3. 隨後，所有MACsec安全幀都使用對稱密碼學(SAC)加密。已建立單獨的TX和RX安全通道。但加密和解密使用相同的金鑰SAK。
4. 當在多接入LAN中檢測到新裝置時（通過EAPOL-MKA消息），金鑰伺服器生成將由所有裝置使用的新金鑰。新金鑰在所有裝置確認後開始使用（請參閱IEEE Std 802.1X-2010的9.17.2部分）。



MACsec資料包

控制幀(EAPOL-MKA)

- EAPOL目的地MAC = 01:80:C2:00:00:03將封包多點傳送至多個目的地
- EAPOL乙太網型別= 0x888E

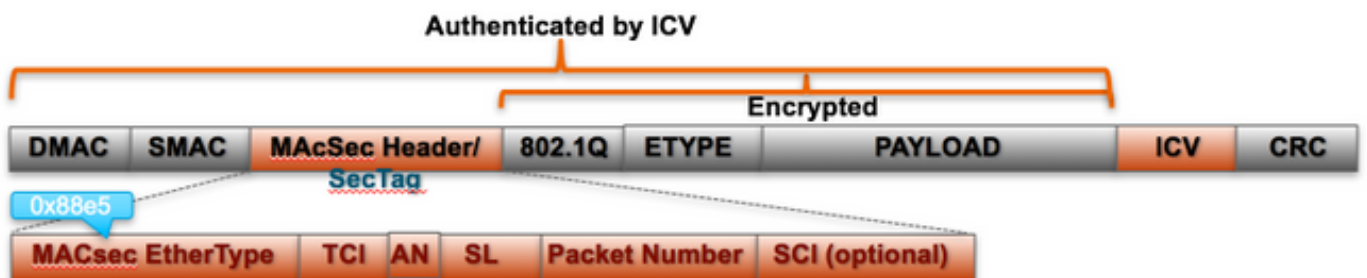
控制幀格式的L2負載。

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

資料幀

MACsec在資料幀上插入兩個附加標籤，最大開銷為32位元組（最小16位元組）。

- SecTag = 8到16位元組（8位元組SCI是可選的）
- ICV = 8到16位元組（基於密碼套）(AES128/256)

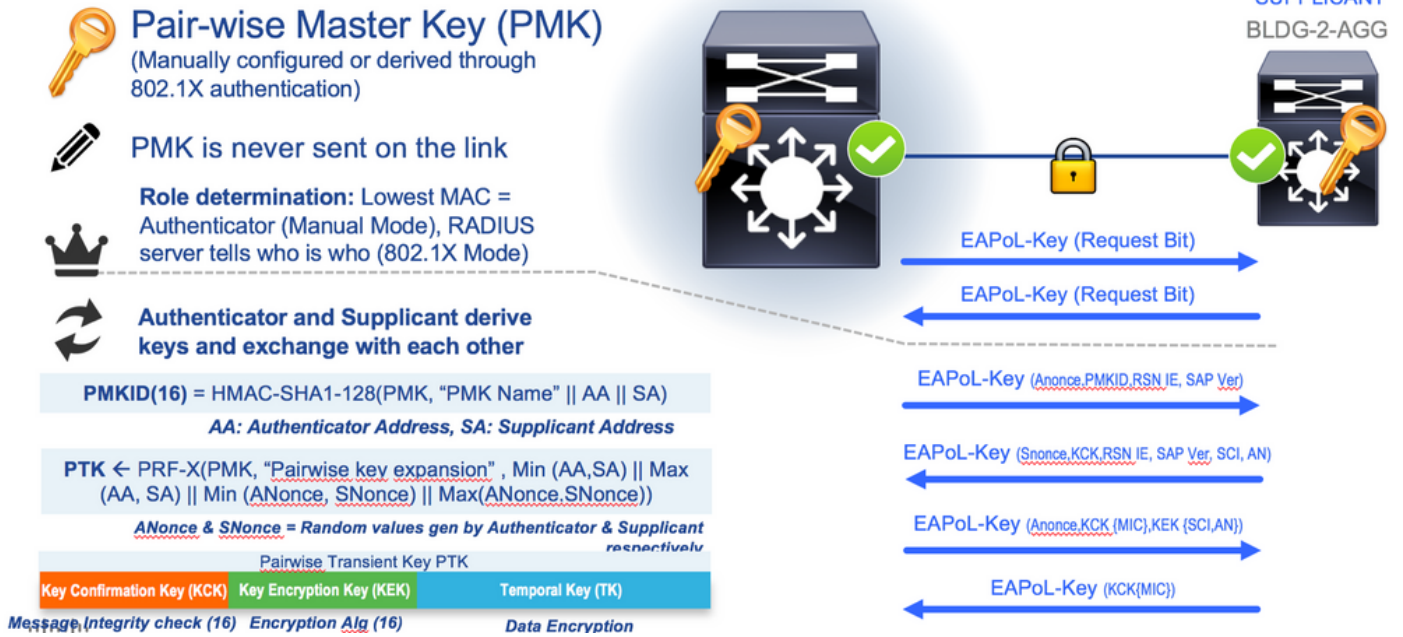


MACsec Tag Format

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)

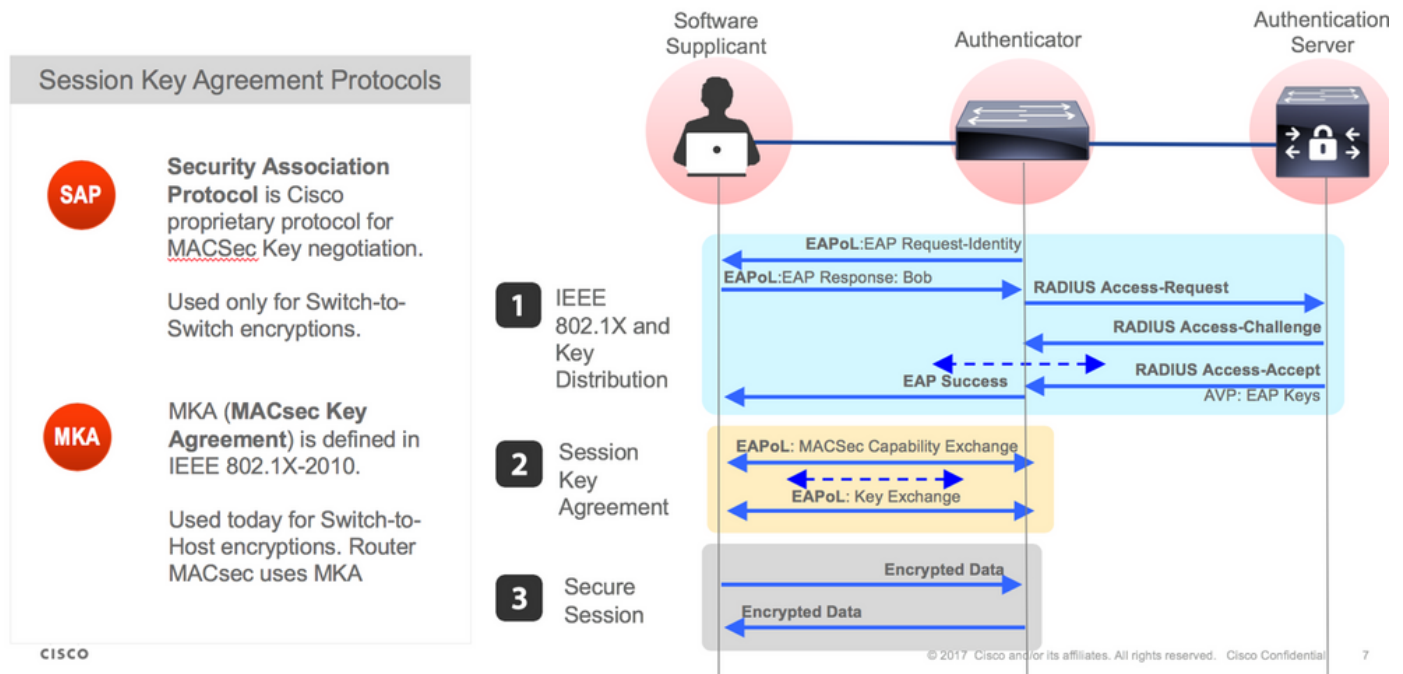
SAP協商

SAP Negotiation

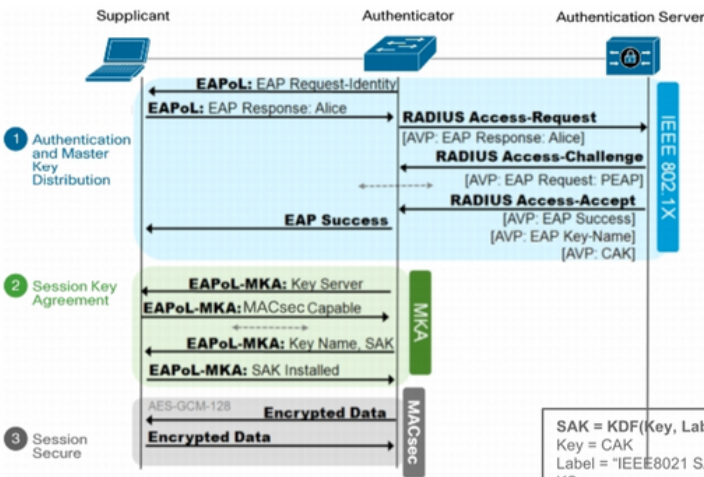


金鑰交換

MACsec Key Derivation Schemes



MKA Exchange



A pairwise CAK (Connectivity Association Key) is derived directly from the EAP MSK:
CAK = KDF(Key, Label, mac1 | mac2, CAKlength)

Key = MSK[0-15] for a 128 bit CAK, MSK[0-31] for a 256 bit CAK
 Label = "IEEE8021 EAP CAK"
 mac1 = the lesser of the two source MAC addr used in the EAPoL-EAP exchange
 mac2 = the greater of the two source MAC addr used in the EAPoL-EAP exchange
 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 CAK using the following transform:
KEK = KDF(Key, Label, Keyid, KEKLength)

Key = CAK
 Label = "IEEE8021 KEK"
 Keyid = the first 16 octets of the CKN, with null octets appended to pad to 16 octets
 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

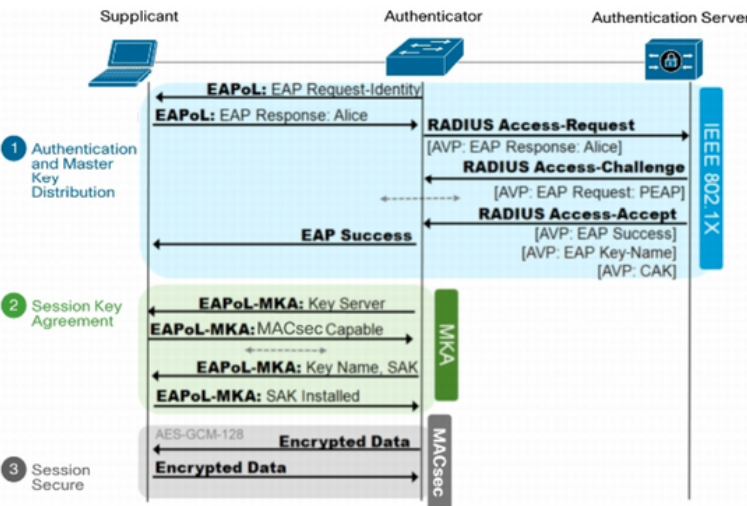
The ICK (ICV Key) is derived from the CAK using the following transform:
ICK = KDF(Key, Label, Keyid, ICKLength)

Key = CAK
 Label = "IEEE8021 ICK"
 Keyid = the first 16 octets of the CKN, with null octets appended to pad to 16
 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 an RNG each time an SAK is generated.
 MI-value list = a concatenation of MI values (in no particular order) from all live participants
 KN = four octets, the Key Number assigned by the Key Server as part of the KI
 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 key Exchange uses:**

- * 802.1x EAP-TLS
- * Pre Shared key (PSK) framework

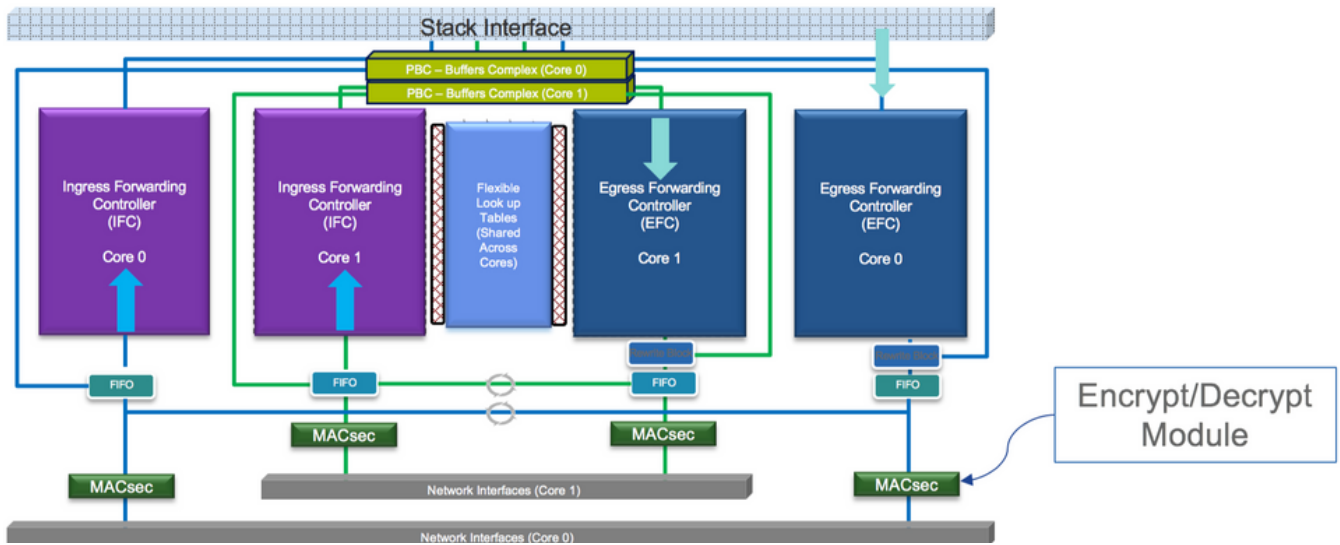
 **MKA 802.1x EAP-TLS**

- * Require Certificate Authority
- * ISE 2.0 +
- * 802.1x AAA config

平台上的MACsec

Where is MACsec performed in Hardware?

Applicable for UADP 2.0/3.0/Mini ASIC



產品相容性矩陣

LAN MACsec Support per Platform

	MACsec	Cat 9200		Cat 9300		Cat 9400		Cat 9500		Cat 9500H / 9600	
		SW	License	SW	License	SW	License	SW	License	SW	License
Switch to Switch	128 Bits SAP	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +	NE	16.9.1 + / 16.11.1 +	NE
	128 Bits MKA	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +	NE	16.9.1 + / 16.11.1 +	NE
	256 Bits MKA	Not Supported		16.6.1 +	NA	16.10.1 +	NA	16.6.1 +	NA	16.9.1 + / 16.11.1 +	NA
	ClearTag Pass Through	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +	NE	16.10.1 + / 16.11.1 +	NE
Host to Switch	128 Bits MKA	16.10.1 +	NE	16.8.1 +	NE	16.9.1 +	NE	16.8.1 +	NE	16.9.1 + / 16.11.1 +	NE
	256 Bits MKA	Not Supported		16.9.1 +	NA	16.10.1 +	NA	16.9.1 +	NA	16.9.1 + / 16.11.1 +	NA

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	Cat 9500H / 9600
Switch to Switch	128 Bits SAP	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	Line Rate	Line Rate	Line Rate
Host to Switch	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	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 Gbps

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

相關資訊

[安全配置指南，Cisco IOS® XE直布羅陀版16.12.x \(Catalyst 9300交換機 \)](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。