

Solucionar problemas do MACSEC no Catalyst 9000

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Vantagens do MacSec](#)

[MACsec e MTU](#)

[Onde o MACsec é usado](#)

[Terminologia](#)

[Cenário 1: segurança de link de switch para switch MACsec com SAP no modo de chave pré-compartilhada \(PSK\)](#)

[Topologia](#)

[Cenário 2: segurança de link de switch a switch MACsec com MKA no modo de chave pré-compartilhada \(PSK\)](#)

[Topologia](#)

[Exemplo de problema de preenchimento](#)

[Outras opções de configuração](#)

[Segurança de Enlace Switch a Switch MACsec com MKA na interface de Pacote/Canal de Porta](#)

[MACsec Switch-to-Switch Link Security em switches intermediários L2, modo PSK](#)

[Restrições](#)

[Informações operacionais do MACsec](#)

[Sequência de Operação](#)

[Pacotes MACsec](#)

[Negociação SAP](#)

[Intercâmbio de chave](#)

[MACsec na plataforma](#)

[Matriz de compatibilidade do produto](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve o recurso MACsec, seus casos de uso e como solucionar problemas do recurso nos switches Catalyst 9000. O escopo deste documento é o MACsec na LAN, entre dois switches/roteadores.

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

- C9300
- C9400

- C9500
- C9600

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Observação: consulte o guia de configuração apropriado para obter os comandos que são usados para habilitar esses recursos em outras plataformas Cisco.

Informações de Apoio

A comunicação de dados em texto claro é susceptível a ameaças à segurança. As violações de segurança podem ocorrer em qualquer camada do modelo OSI. Algumas das violações comuns na camada 2 são farejamento, interceptação de pacotes, violação, injeção, falsificação de endereço MAC, falsificação ARP, ataques de negação de serviço (DoS) contra um servidor DHCP e saltos de VLAN.

O MacSec é uma tecnologia de criptografia L2 descrita no padrão IEEE 802.1AE. O MACsec protege os dados em meios físicos e torna impossível que os dados sejam comprometidos em camadas superiores. Como resultado, a criptografia MACsec tem prioridade sobre qualquer outro método de criptografia para camadas superiores, como IPsec e SSL.

Vantagens do MacSec

Modo orientado ao cliente: o MACsec é usado em configurações nas quais dois switches que estão fazendo peering entre si podem alternar como um servidor de chave ou um cliente de chave antes de trocar chaves. O servidor de chaves gera e mantém o CAK entre os dois pares.

Verificação de integridade de dados: o MACsec usa o MKA para gerar um valor de verificação de integridade (ICV) para o quadro que chega à porta. Se o ICV gerado for o mesmo que o ICV no quadro, o quadro será aceito; caso contrário, ele será descartado.

Criptografia de dados: o MACsec fornece criptografia em nível de porta nas interfaces dos switches. Isso significa que os quadros enviados da porta configurada são criptografados e os quadros recebidos na porta são descriptografados. O MACsec também fornece um mecanismo no qual você pode configurar se somente quadros criptografados ou todos

quadros (criptografados e simples) são aceitos na interface.

Proteção contra repetição: quando os quadros são transmitidos através da rede, há uma possibilidade de que os quadros saiam da sequência ordenada. O MACsec fornece uma janela configurável que aceita um número especificado de quadros fora de sequência.

MACsec e MTU

O cabeçalho MACsec adiciona até 32 bytes de sobrecarga de cabeçalho. Considere um MTU de sistema/interface maior nos switches no caminho para considerar a sobrecarga adicional adicionada pelo cabeçalho MACsec. Se a MTU for muito baixa, você poderá ver perda/atraso de pacote inesperado para aplicativos que precisam usar uma MTU mais alta.

Observação: se houver um problema relacionado ao MACSEC, verifique se o GBIC em ambas as

extremidades é suportado pela [Matriz de Compatibilidade](#) .

Onde o MACsec é usado

Casos de uso do campus

- Host para switch
- Entre locais ou edifícios
- Entre andares em uma multilocação

Casos de uso de data center

- Interconexão de data center
- Servidor para switch

Casos de uso de WAN

- Interconexão de data center
- Interconexão do campus
- Hub-Spoke

Terminologia

MKA	Contrato de Chave MACsec	definido no IEEE 802.1X REV-2010 como um protocolo de acordo-chave para descobrir pares MACsec e chaves de negociação
CAK	Chave de Associação de Conectividade	chave mestra de longa duração usada para gerar todas as outras chaves usadas para MACsec. As implementações de LAN derivam isso do MSK (gerado durante a troca de EAP)
PMK	Chave mestra par a par	Um dos componentes usados para derivar as chaves de sessão usadas para criptografar o tráfego. Configurado manualmente ou derivado de 802.1X
CKN	nome da chave CAK	usado para configurar o valor de chave ou CAK. Apenas um número par de <u>caracteres HEX</u> é permitido até 64 caracteres.
SAK	Chave de Associação Segura	derivado pelo servidor de chave selecionado do CAK e é a chave usada pelo roteador/dispositivos finais para criptografar o tráfego para uma determinada sessão.
ICV	Chave de valor de verificação de integridade	derivado de CAK e marcado em cada quadro de dados/controlado para comprovar que o quadro é de um peer autorizado. 8 a 16 bytes dependendo do conjunto de cifras
KEK	Chave de criptografia de chave	derivado de CAK (a chave pré-compartilhada) e usado para proteger as chaves MacSec
SIC	Identificador de canal seguro	Cada porta virtual recebe um identificador de canal seguro (SCI) exclusivo com base no endereço MAC da interface física concatenada com um ID de porta de 16 bits

Cenário 1: segurança de link de switch para switch MACsec com SAP no modo de chave pré-compartilhada (PSK)

Topologia


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

Etapa 2. Verifique o estado do MACsec e se os parâmetros/contadores estão corretos

```
<#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

Etapa 3. Revise as depurações de software quando o link for ativado.

<#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
```

Etapa 4. Revisar rastreamentos no nível da plataforma quando o link for ativado

```
<#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_tx
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_fra
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

```

Etapa 5. Verificar o estado da interface MACsec no 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
```

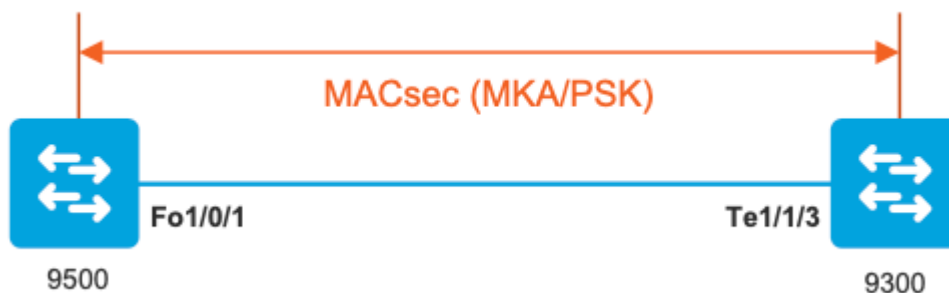
```
LEAD_PORT_EGRESS_MAC_SEC_SUB_PORT value 0 Pass
```

```
LEAD_PORT_EGRESS_MACSEC_ENCRYPTED value 0 Pass
```

snip

Cenário 2: segurança de link de switch a switch MACsec com MKA no modo de chave pré-compartilhada (PSK)

Topologia



Etapa 1. Validar a configuração em ambos os lados do 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
```

Etapa 2. Validar se o MACsec está habilitado e se todos os parâmetros/contadores estão corretos

```
<#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


```

conf t
  key chain macsec1 macsec

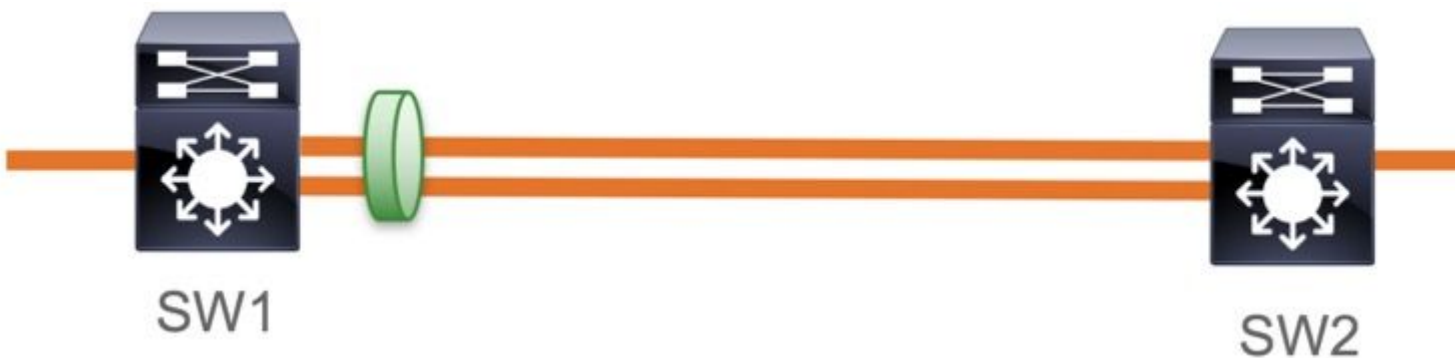
key 01 --> Device does automatic padding.

  key-octet-string 12345678901234567890123456789012
end

```

Outras opções de configuração

Segurança de Enlace Switch a Switch MACsec com MKA na interface de Pacote/Canal de Porta



- Canais de porta L3 e L2 (LACP, PAgP e Modo ON)
- Tipos de criptografia (AES-128 e AES-256 (AES-256 é aplicável para a licença Advantage))
- Troca de chaves MKA PSK apenas

Plataformas suportadas:

- Catalyst 9200 (somente AES-128)
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500 e Catalyst 9500H
- Catalyst 9600

Exemplo de Configuração de Switch para Switch Etherchannel

A configuração da cadeia de chaves e da política MKA permanece a mesma, conforme mostrado anteriormente na seção de configuração MKA.

```
<#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
```

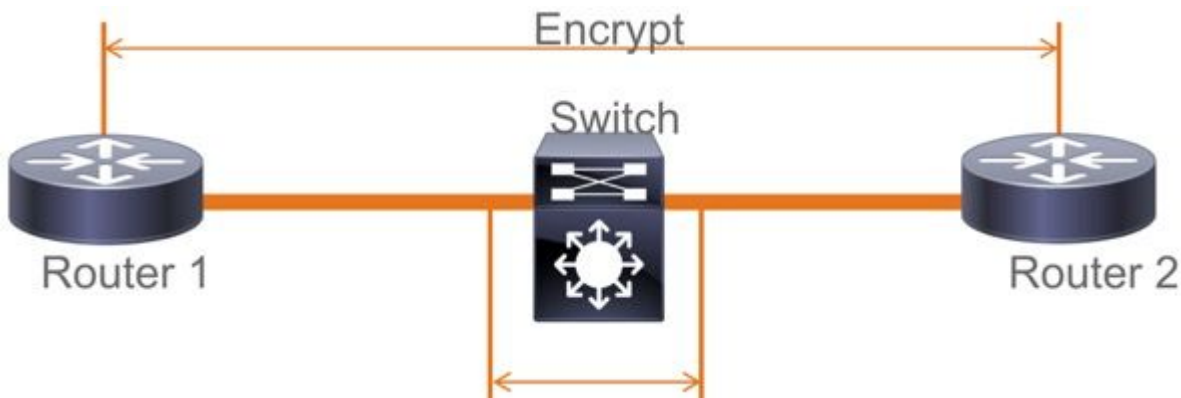
MACsec Switch-to-Switch Link Security em switches intermediários L2, modo PSK

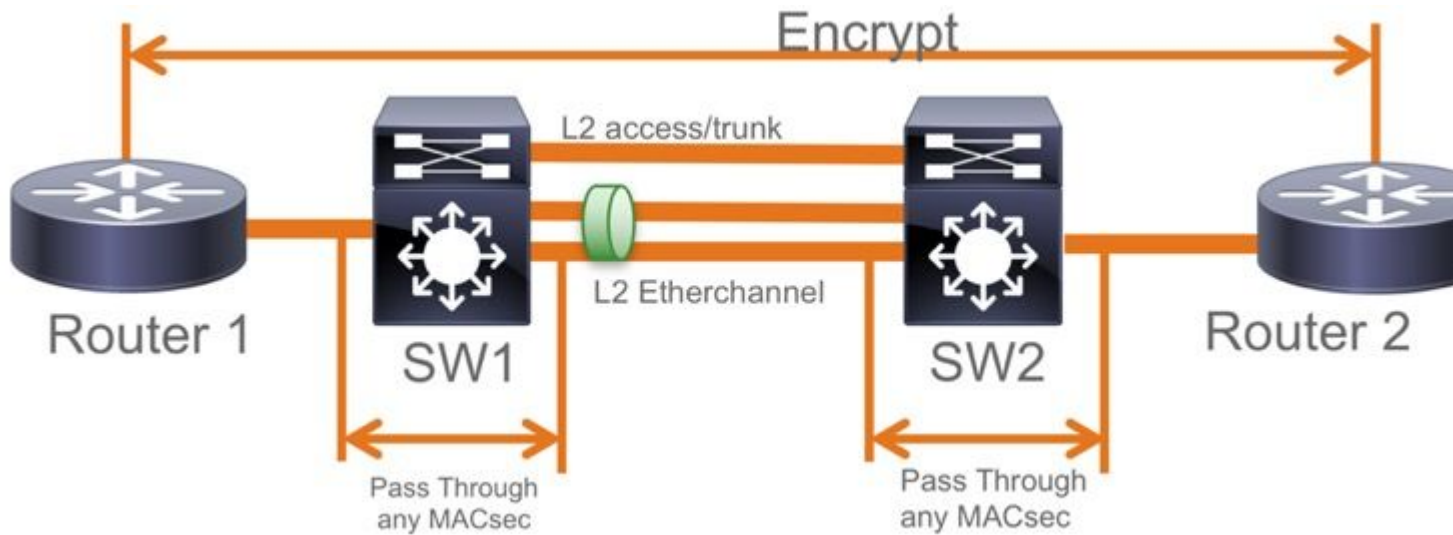
Esta seção aborda alguns dos cenários de WAN MACsec suportados em que o Cat9K precisa passar pacotes criptografados de forma transparente.

Há casos em que os roteadores não estão diretamente conectados, mas têm switches intermediários L2, e os switches L2 devem ignorar os pacotes criptografados sem nenhum processamento da criptografia.

Os switches Catalyst 9000 encaminham pacotes de forma transparente com Clear Tag a partir da versão 16.10(1)

- A passagem é compatível com MKA/SAP
- Suportado em acesso L2, tronco ou Etherchannels
- Suportado por padrão (sem CLIs de configuração para ativar/desativar)
- **Certifique-se de que os roteadores enviem quadros EAPOL com tipo de éter não padrão (0x888E)**

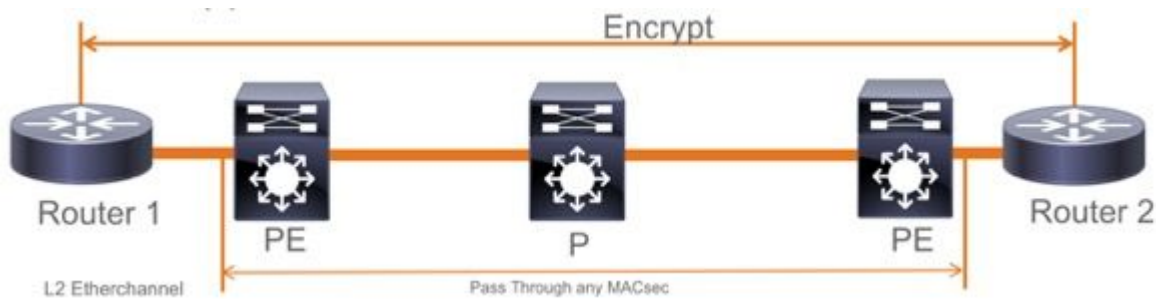




Topologia EoMPLS / VPLS

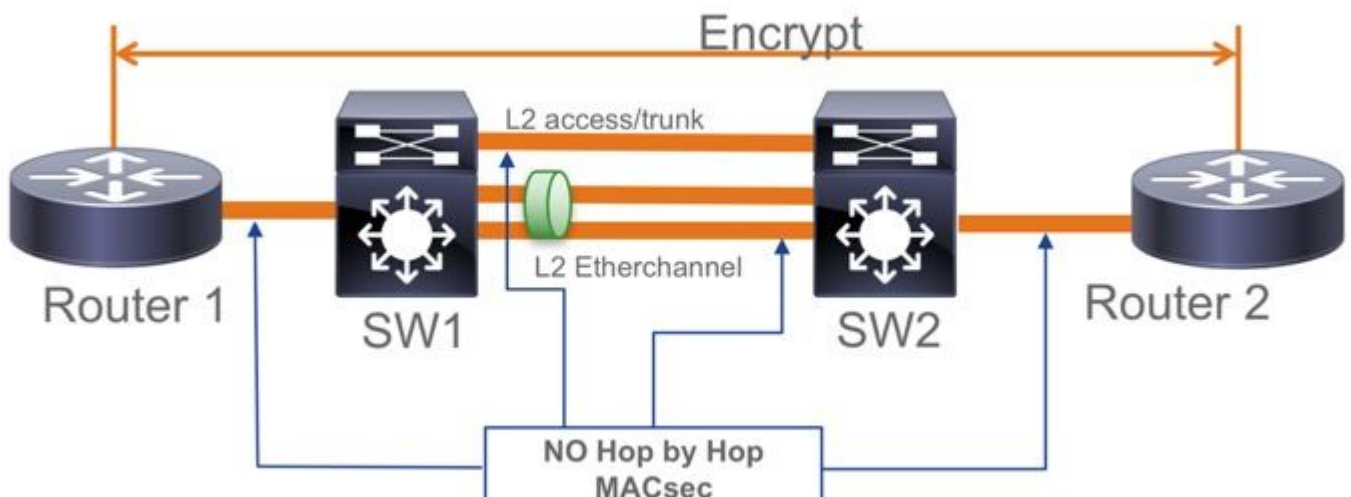
Plataformas suportadas Cat 9300/9400,9500/9500H como dispositivos "PE" ou "P"

- VPLS
- EoMPLS
- Suportado por padrão (sem CLIs de configuração para ativar/desativar)
- Iniciar 16.10(1)

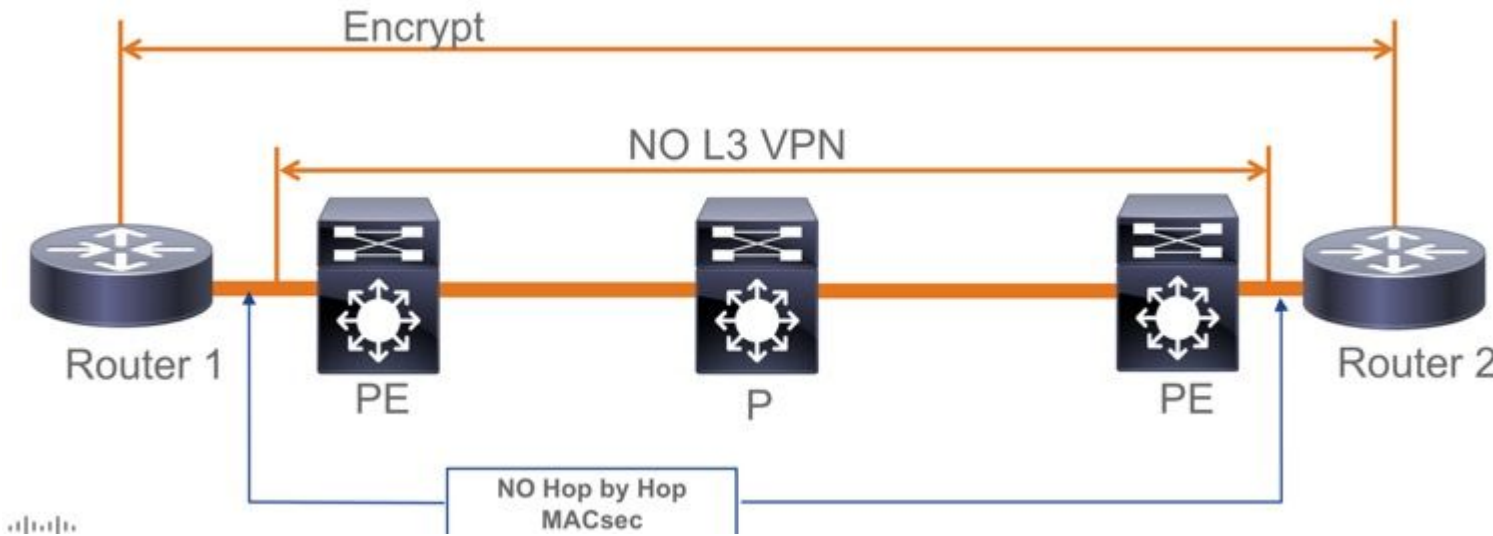


Restrições

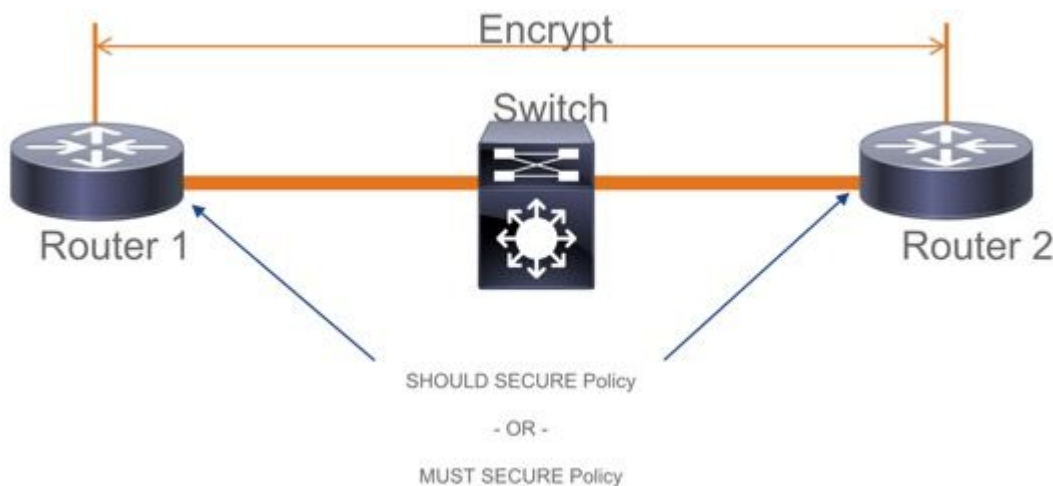
Não há suporte para criptografia dupla. O MACsec de ponta a ponta com marca Clear requer que os switches Hop by Hop não sejam habilitados nos links L2 conectados diretamente



- ClearTag + EoMPLS com switches intermediários somente de Camada 2, MACsec não pode ativar no link CE-PE
- ClearTag + L3VPN com switches intermediários não suportados



- Não há suporte para "Should Secure" no modo PSK, "Must Secure" é o modo padrão
- A política Must Secure não criptografa somente EAPoL para negociar as configurações MACsec

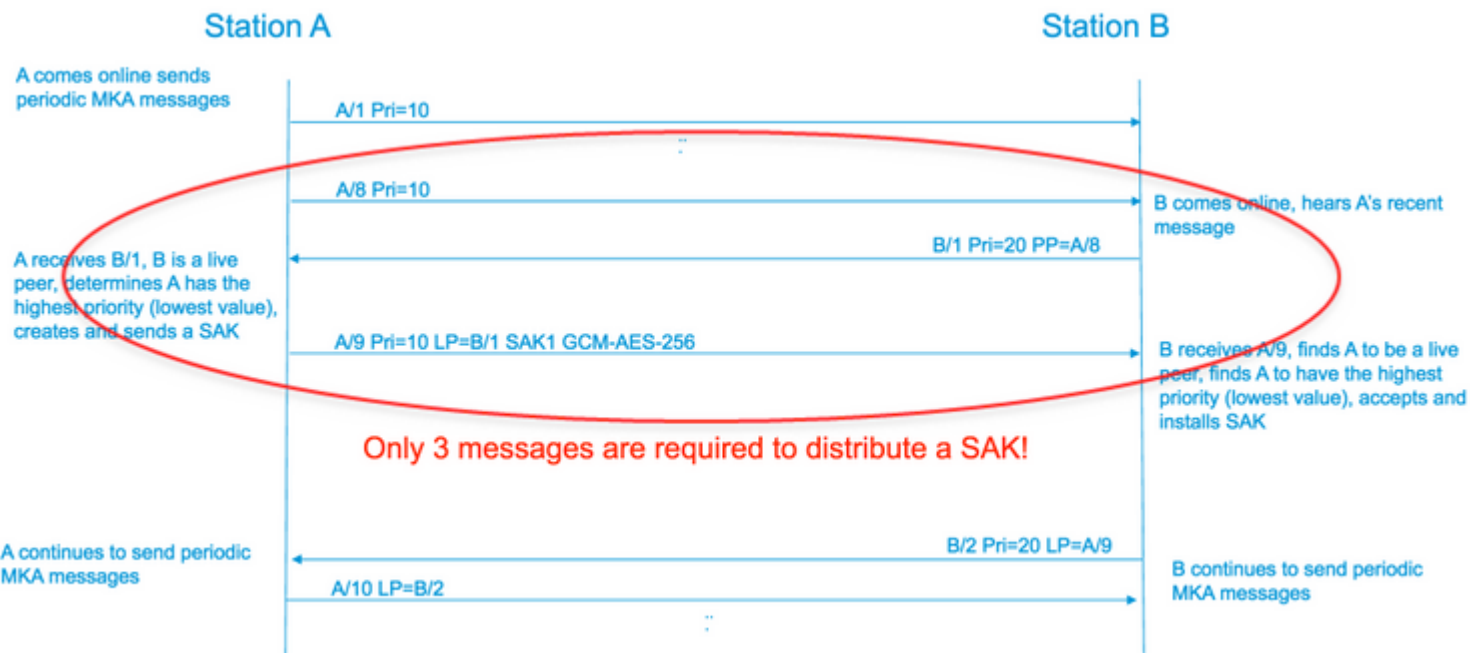


Informações operacionais do MACsec

Sequência de Operação

1. Quando o link e ambos os dispositivos finais surgem, eles trocam quadros MKA (**ethertype = 0x888E**, o mesmo que EAPoL com tipo de pacote como MKA). É um protocolo de negociação multiponto a multiponto. O valor da chave CAK (normalmente pré-compartilhada estática), o nome da chave (CKN) devem corresponder e o ICV deve ser válido para que os pares sejam descobertos e aceitos.
2. O dispositivo com a prioridade mais baixa do servidor de chaves (padrão = 0) é escolhido como o servidor de chaves. O servidor de chaves gera o SAK e o distribui através de mensagens MKA. Em caso de empate, o valor mais elevado do SIC (identificador de canal seguro) ganha.
3. Subsequentemente, todos os quadros protegidos do MacSec são criptografados com o SAK (criptografia simétrica). Há canais seguros TX e RX separados criados. Mas a mesma chave SAK é usada para criptografar e descriptografar.
4. Quando um novo dispositivo é detectado em uma LAN multiacesso (através de mensagens EAPoL-

MKA), o servidor de chaves gera uma nova chave a ser usada por todos os dispositivos. A nova chave entra em uso depois de ser reconhecida por todos os dispositivos (consulte a seção 9.17.2 do padrão IEEE 802.1X-2010).



Pacotes MACsec

Quadro de controlo (EAPOL-MKA)

- MAC de destino EAPOL = 01:80:C2:00:00:03 para enviar os pacotes por multicast para vários destinos
- Tipo de éter EAPOL = 0x888E

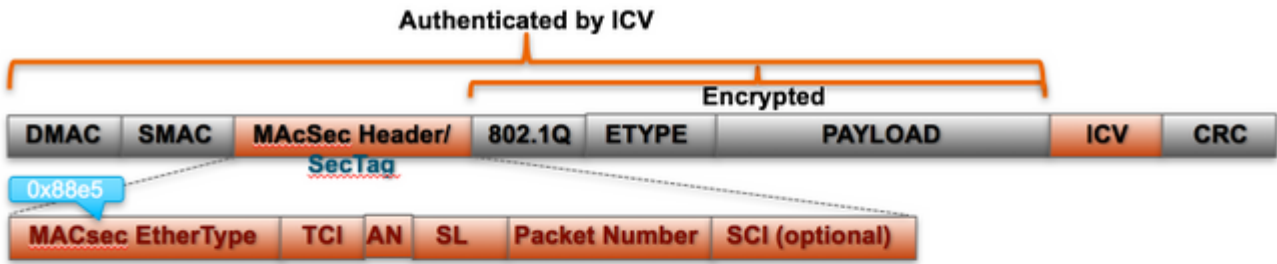
Carga útil de L2 no formato do quadro de Controle

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

Quadro de dados

O MACSec insere duas tags adicionais nos quadros de dados com sobrecarga máxima de **32 bytes** (mínimo de 16 bytes).

- **SecTag** = 8 a 16 bytes (SCI de 8 bytes é opcional)
- **ICV** = 8 a 16 bytes com base no naipe de cifra (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)

Negociação SAP

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 (

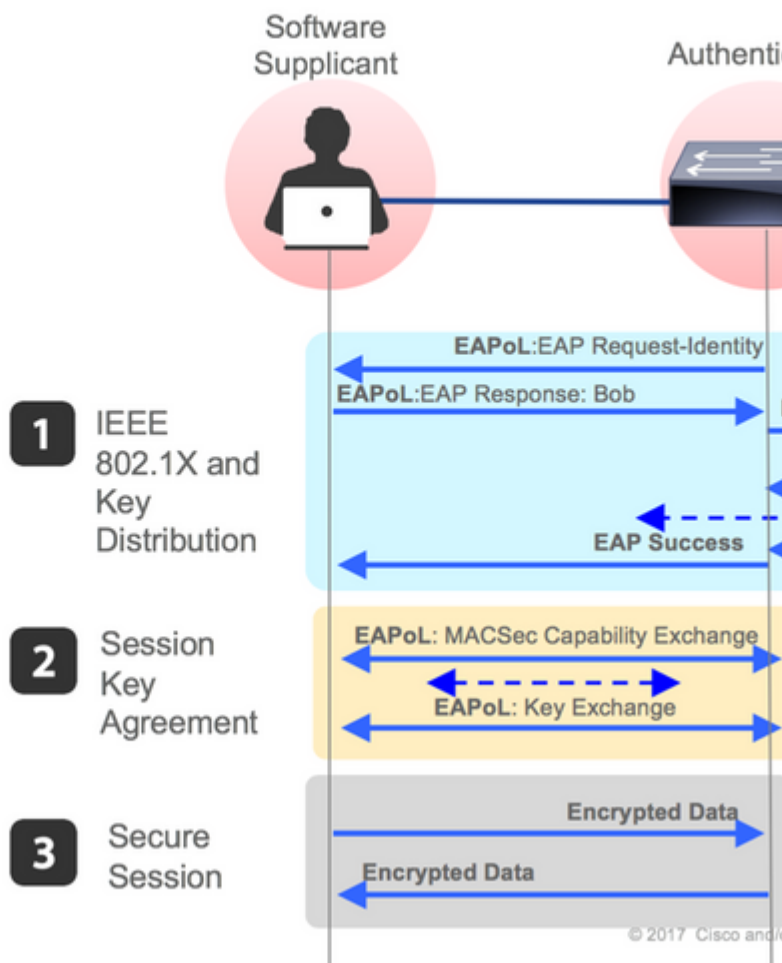
EAPoL-

MACsec Key Derivation Schemes

Session Key Agreement Protocols

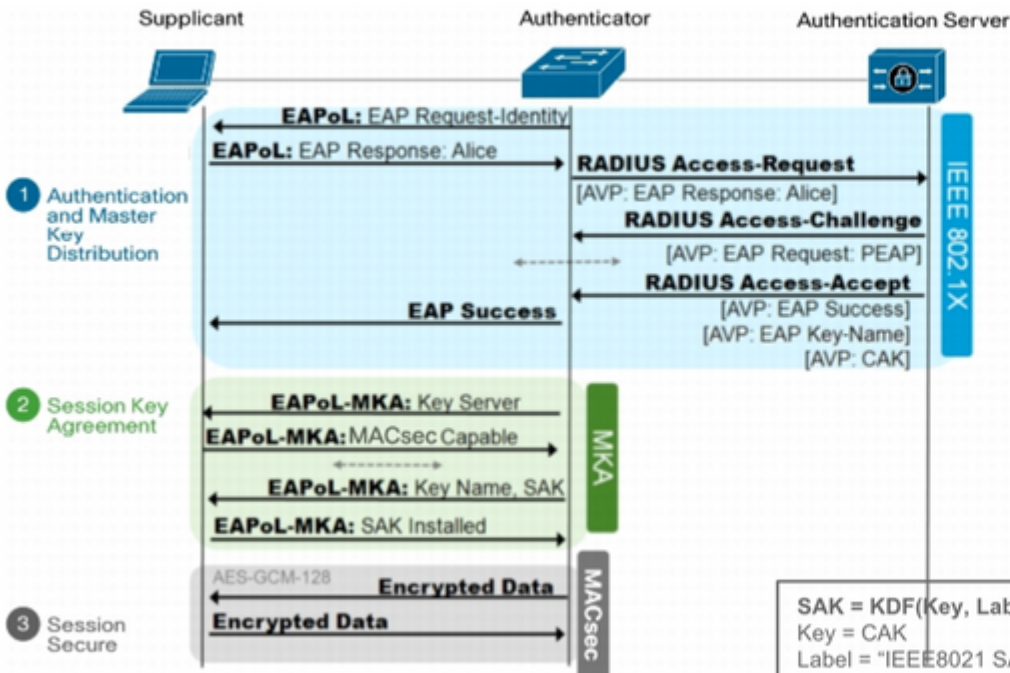
- SAP** **Security Association Protocol** is Cisco proprietary protocol for MACSec Key negotiation. Used only for Switch-to-Switch encryptions.
- MKA** **MKA (MACsec Key Agreement)** is defined in IEEE 802.1X-2010. Used today for Switch-to-Host encryptions. Router MACsec uses MKA

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)

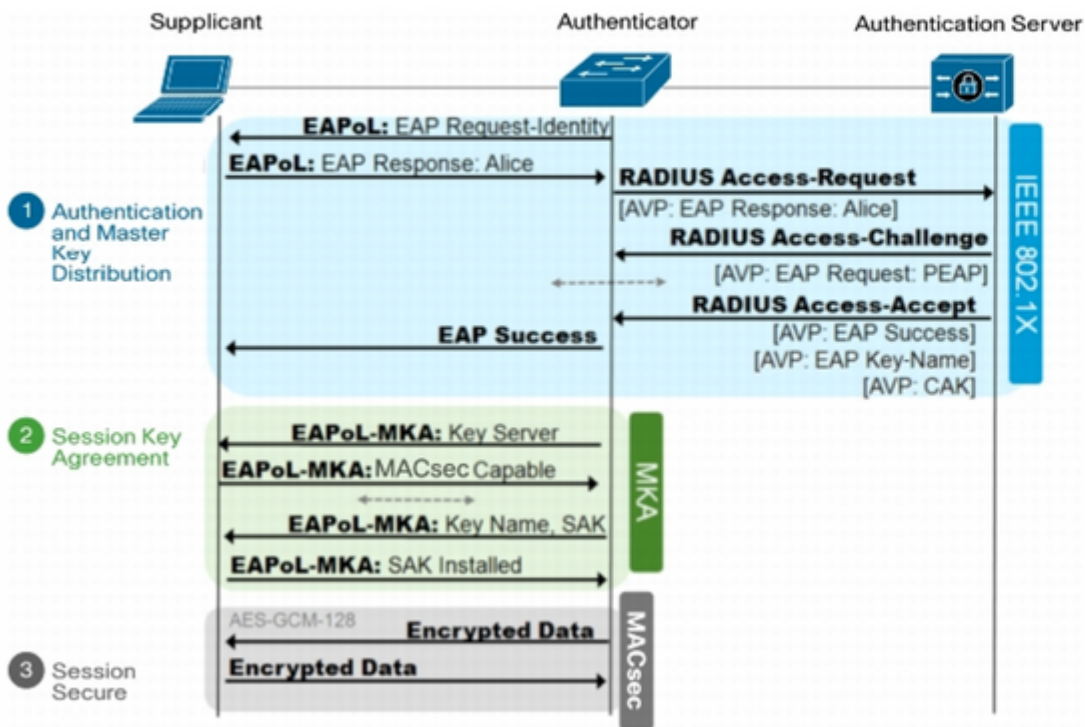
Key = CAK
 Label = "IEEE8021 ICK"
 Keyid = the first 16 octets of the CKN, with the most significant octet first
 ICKLength = two octets representing an integer value (128 for a 128 bit ICK, 256 for a 256 bit ICK) with the most significant octet first.

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

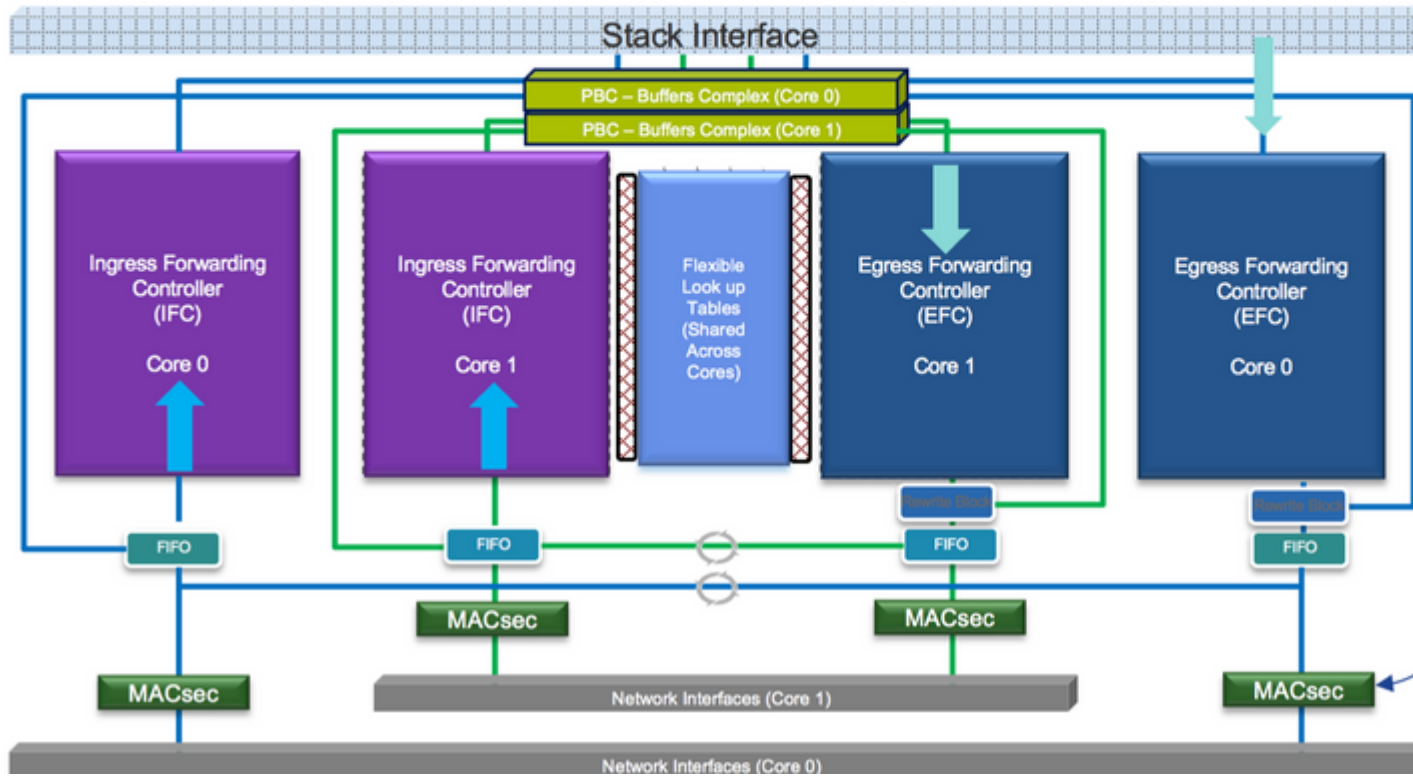


MKA
* Rec
* ISE
* 802

MACsec na platforma

Where is MACsec performed in Hardware?

Applicable for UADP 2.0/3.0/Mini ASIC



Matriz de compatibilidade do produto

LAN MACsec Support per Platform

	MACsec	Cat 9200		Cat 9300		Cat 9400		Cat 9500
		SW	License	SW	License	SW	License	SW
Switch to Switch	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 +
Host to Switch	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

Informações Relacionadas

[Guia de configuração de segurança, Cisco IOS XE Gibraltar 16.12.x \(switches Catalyst 9300\)](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.