Fehlerbehebung bei MACSEC auf Catalyst 9000

Inhalt

Einleitung

In diesem Dokument werden die MACsec-Funktion, ihre Anwendungsfälle und die Fehlerbehebung für die Funktion der Catalyst 9000-Switches beschrieben. Der Umfang dieses Dokuments umfasst MACsec im LAN zwischen zwei Switches/Routern.

Voraussetzungen

Anforderungen

Es gibt keine spezifischen Anforderungen für dieses Dokument.

Verwendete Komponenten

- C9300
- C9400
- C9500
- C9600

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

> **Hinweis**: Informationen zu den Befehlen, die zur Aktivierung dieser Funktionen auf anderen Cisco Plattformen verwendet werden, finden Sie im entsprechenden Konfigurationsleitfaden.

Hintergrundinformationen

Die Textkommunikation ist anfällig für Sicherheitsbedrohungen. Sicherheitslücken können auf jeder Ebene des OSI-Modells auftreten. Zu den häufigsten Sicherheitsverletzungen auf Layer 2 gehören Sniffing, Paketabhören, Manipulation, Einschleusung, MAC-Adressen-Spoofing, ARP-Spoofing, DoS-Angriffe (Denial of Service) auf einen DHCP-Server und VLAN-Hopping.

MacSec ist eine L2-Verschlüsselungstechnologie, die im IEEE 802.1AE-Standard beschrieben wird. MACsec schützt die Daten auf physischen Medien und verhindert, dass Daten auf höheren Ebenen kompromittiert werden. Daher hat die MACsec-Verschlüsselung für höhere Schichten wie IPsec und SSL Vorrang vor allen anderen Verschlüsselungsmethoden.

Vorteile von MacSec

Client-Oriented Mode (Client-orientierter Modus): MACsec wird in Konfigurationen verwendet, in denen zwei Switches, die miteinander als Peering verbunden sind, vor dem Austausch von Schlüsseln als Schlüsselserver oder Schlüsselclient abwechseln können. Der Schlüsselserver generiert und verwaltet die CAK zwischen den beiden Peers.

Datenintegritätsprüfung: MACsec verwendet MKA, um einen Integritätsprüfwert (ICV) für den Frame zu generieren, der am Port ankommt. Wenn der generierte ICV mit dem ICV im Frame übereinstimmt, wird der Frame akzeptiert, andernfalls wird er verworfen.

Datenverschlüsselung: MACsec bietet Verschlüsselung auf Port-Ebene an den Schnittstellen von Switches. Das bedeutet, dass die über den konfigurierten Port gesendeten Frames verschlüsselt und die über den Port empfangenen Frames entschlüsselt werden. MACsec bietet auch einen Mechanismus, mit dem Sie konfigurieren können, ob nur verschlüsselte Frames oder alle

-Frames (verschlüsselt und unverschlüsselt) auf der Schnittstelle akzeptiert werden.

Wiedergabeschutz: Wenn Frames über das Netzwerk übertragen werden, besteht die Möglichkeit, dass die Frames die geordnete Sequenz verlassen. MACsec stellt ein konfigurierbares Fenster bereit, das eine angegebene Anzahl von Frames akzeptiert, die nicht der Sequenz entsprechen.

MACsec und MTU

Der MACsec-Header summiert bis zu 32 Byte Header-Overhead. Stellen Sie sich eine größere System/Schnittstellen-MTU auf Switches im Pfad vor, um den zusätzlichen Overhead zu berücksichtigen, der durch den MACsec-Header hinzugefügt wird. Wenn die MTU zu niedrig ist, kann es bei Anwendungen, die eine höhere MTU benötigen, zu unerwarteten Paketverlusten/Verzögerungen kommen.

> **Hinweis**: Wenn ein Problem mit MACSEC auftritt, stellen Sie sicher, dass der GBIC an beiden Enden gemäß der Kompatibilitätsmatrix unterstützt wird.

Wo MACsec verwendet wird

Anwendungsfälle in Campus-Umgebungen

- Host-zu-Switch
- Zwischen Standorten oder Gebäuden
- Zwischen Etagen in einer Multi-Tenant-Umgebung

Anwendungsfälle im Rechenzentrum

- Data Center Interconnect
- Server-zu-Switch

WAN-Anwendungsfälle

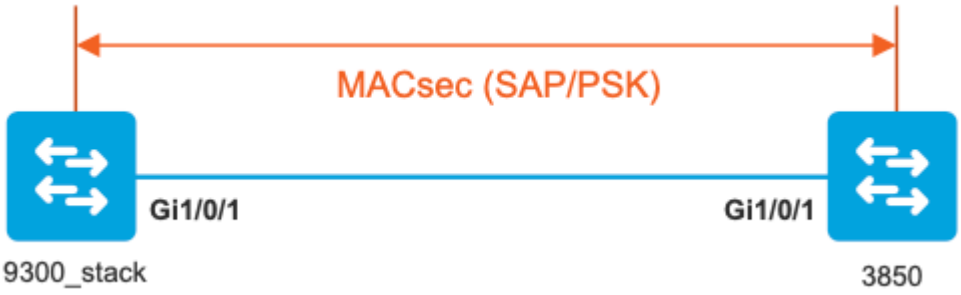
- Data Center Interconnect
- Campus-Verbindung
- Hub-Spoke

Terminologie

MKA	MACsec-Schlüsselvereinbarung	definiert in IEEE 802.1X REV-2010 als Key Agreement-Protokoll zur Erkennung von MACsec-Peers und zur Aushandlung von Schlüsseln
KUCHEN	Verbindungszuordnungsschlüssel	Langlebiger Hauptschlüssel zum Generieren aller anderen Schlüssel, die für MACsec verwendet werden. LAN-Implementierungen leiten dies von MSK ab (wird beim EAP-Austausch generiert).
PMK	PAARWEISE STASTSCHLÜSSEL	Eine der zur Ableitung der Sitzungsschlüssel verwendeten Komponenten, die zur Verschlüsselung des Datenverkehrs verwendet werden. Manuell konfiguriert oder abgeleitet von 802.1X
CKN	CAK-Schlüsselname	wird verwendet, um den Schlüsselwert oder die CAK zu konfigurieren. Nur eine gerade Anzahl von Hexadezimalziffern mit bis zu 64 Zeichen ist zulässig.
SAK	Sicherer Zuordnungsschlüssel	wird vom ausgewählten Schlüsselserver aus der CAK abgeleitet und ist der Schlüssel, der vom Router bzw. den Endgeräten zum Verschlüsseln des Datenverkehrs für eine bestimmte Sitzung verwendet wird.
ICV	Integritätsprüfungswertschlüssel	aus CAK abgeleitet und in jedem Daten-/Steuerungs-Frame mit Tags versehen wird, um nachzuweisen, dass der Frame von einem autorisierten Peer stammt. 8-16 Bytes je Verschlüsselungssuite
TASK	Schlüsselverschlüsselungsschlüssel	abgeleitet von CAK (dem vorinstallierten Schlüssel) und zum Schutz der MacSec-Schlüssel verwendet
SCI	Secure Channel Identifier	Jeder virtuelle Port erhält eine eindeutige Secure Channel Identifier (SCI), die auf der MAC-Adresse der physischen Schnittstelle basiert, die mit einer 16-Bit-Port-ID verknüpft ist.

Szenario 1: Sicherheit der MACsec-Switch-to-Switch-Verbindung mit SAP im PSK-Modus (Pre-Shared Key)

Topologie



Schritt 1: Validierung der Konfiguration auf beiden Seiten der Verbindung

<#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

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

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 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA mode-list gcm-encrypt

<--

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

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.

Schritt 2: Überprüfen Sie den MACsec-Status, und die Parameter/Zähler sind richtig.

<#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
```

Schritt 3: Überprüfen Sie die Software-Fehlerbehebungen, wenn der Link angezeigt wird.

<#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 AAAAAAAAAAAAAAAAAAA**

```
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 switc**

***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 inte**

```
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:d0c7.8**

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**

Schritt 4: Überprüfen Sie die Ablaufverfolgungen auf Plattformebene beim Aufrufen des Links.

<#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 R
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 han

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 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 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_t
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_t
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: N

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

Schritt 5: Überprüfen des Status der MACsec-Schnittstelle in der Hardware

<#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-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)

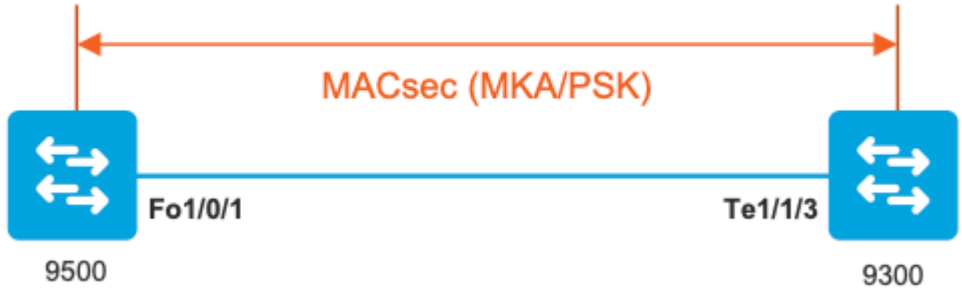
```
LEAD_PORT_PHY_MAC_SEC_SUB_PORT_ENABLED value 0 Pass
LEAD_PORT_SGT_ALLOWED value 0 Pass
```

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**
```

Szenario 2: MACsec-Switch-to-Switch-Verbindungssicherheit mit MKA im PSK-Modus (Pre-Shared Key)

Topologie



Schritt 1: Validierung der Konfiguration auf beiden Seiten der Verbindung

<#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 te1/1/3


interface te1/1/3
macsec network-link


mka policy MKA


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

Schritt 2: Überprüfen, ob MACsec aktiviert ist und alle Parameter/Zähler korrekt sind

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...
==
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   <-- 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

      KS    DP       CO SAKR    ICVIND  Cipher   Interfaces

Name

       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
```

Schritt 3 bis Schritt 5

Befolgen Sie die gleichen Anweisungen wie in Szenario 1.

> **Warnung**: **Zur Interoperabilität**. Bitte beachten Sie, dass einige Plattformen Padding durchführen und einige Plattformen dies nicht tun. Dies kann zu Schlüsselproblemen führen, bei denen die mka-Sitzung im "Init"-Status verbleibt. Sie können dies mit "**show mka sessions" überprüfen**.

Beispiel für Füllungsprobleme

Dieser Anwendungsfall zeigt einen Catalyst 9500 und einen Nexus 7000 in NX-OS 8.2(2), kann aber auch bei Catalyst-Geräten wie C3560CX auftreten.

(Cisco Bug-ID CSCvs92023 dokumentiert das Problem).

- Wenn Sie die Konfiguration aus Szenario 2 befolgen, erstellt MKA den Tunnel aufgrund einer Schlüsselfehlanpassung nicht.
- Sie müssen den Schlüssel manuell mit 0 auf der 9500-Seite ausfüllen, da dieses Gerät kein Padding ausführt.

Catalyst 9500

<#root>

```
conf t
  key chain macsec1 macsec
     key
```

0100000000000000000000000000000000000000000000000000000000000000 --> device does not do padding automati

```
       key-string 12345678901234567890123456789012
       end
```

Nexus 7000

<#root>

```
conf t
  key chain macsec1 macsec
```

key 01 --> Device does automatic padding.

```
      key-octet-string 1234567890123456789012345678901234567890123456789012
      end
```

Weitere Konfigurationsoptionen

MACsec-Switch-to-Switch-Link-Sicherheit mit MKA an gebündelter/Port-Channel-Schnittstelle

- L3- und L2-Port-Channels (LACP, PAgP und Modus EIN)
- Verschlüsselungstypen (AES-128 und AES-256 (AES-256 gilt für Advantage-Lizenz)
- Nur MKA PSK für Key Exchange

Unterstützte Plattformen:

- Catalyst 9200 (nur AES-128)
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500 und Catalyst 9500H
- Catalyst 9600

Switch-to-Switch-EtherChannel-Beispielkonfiguration

Die Konfiguration der Schlüsselbund- und MKA-Richtlinien ist identisch mit der Konfiguration im Abschnitt zur MKA-Konfiguration.

<#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
```

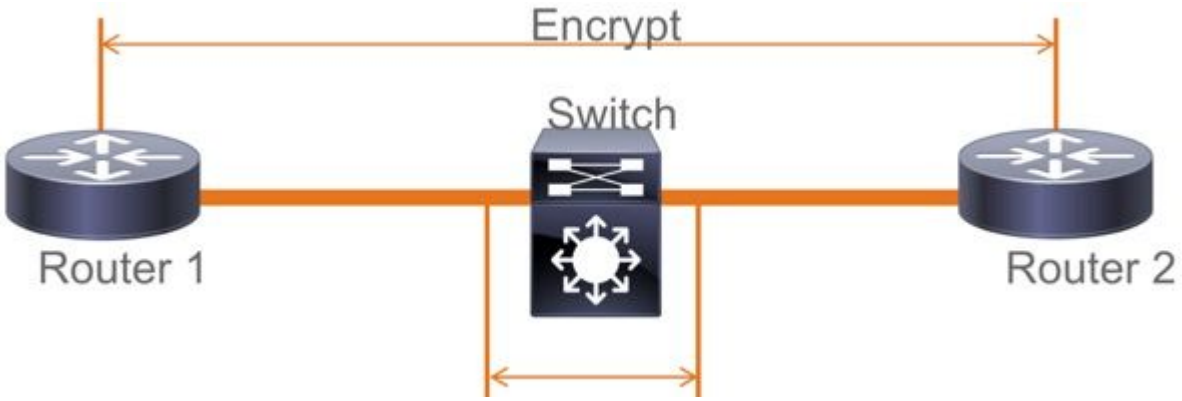
MACsec-Switch-to-Switch-Link-Sicherheit für L2-Zwischenswitches, PSK-Modus

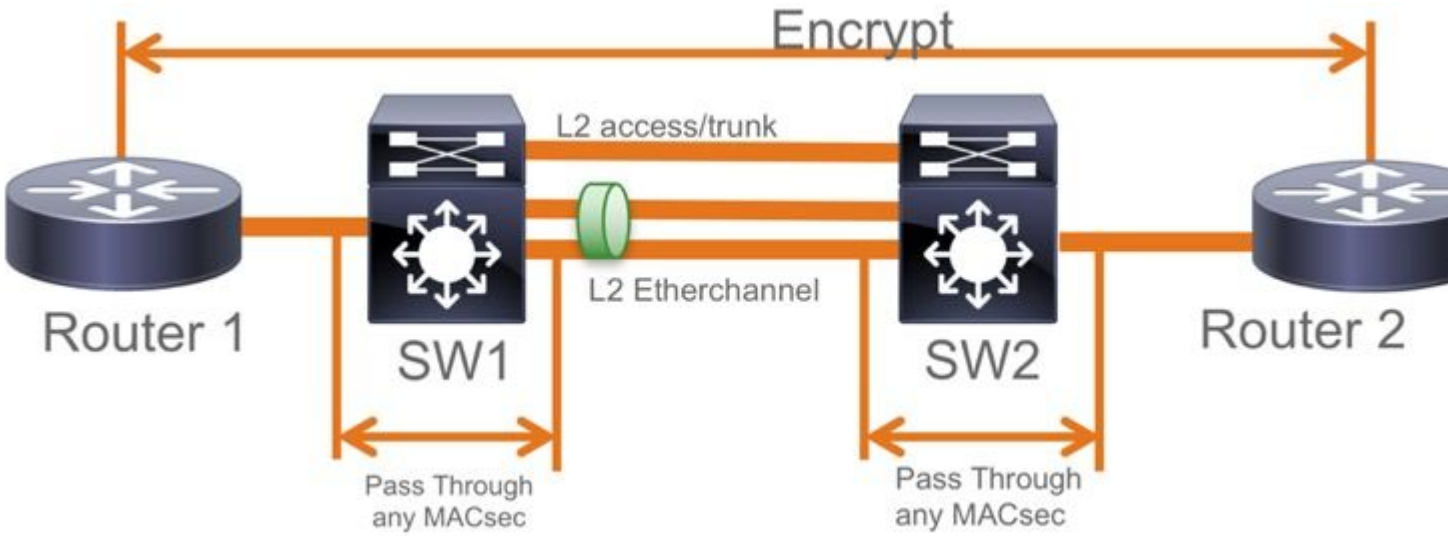
In diesem Abschnitt werden einige der unterstützten WAN-MACsec-Szenarien beschrieben, in denen Cat9K verschlüsselte Pakete transparent weiterleiten muss.

Es gibt Fälle, in denen Router nicht direkt verbunden sind, aber über L2-Switches verfügen, und die L2-Switches sollten die verschlüsselten Pakete umgehen, ohne die Verschlüsselung zu verarbeiten.

Catalyst 9000-Switches leiten transparente Pakete mit Clear Tag weiter, beginnend mit 16.10(1)

- Pass-Through wird für MKA/SAP unterstützt.
- Unterstützt auf L2-Zugang, Trunk oder EtherChannels
- Standardmäßig unterstützt (keine CLIs für Konfiguration zum Aktivieren/Deaktivieren)
- **Sicherstellen, dass Router EAPOL-Frames vom Typ "Ether" (nicht Standard) (0x888E) senden**

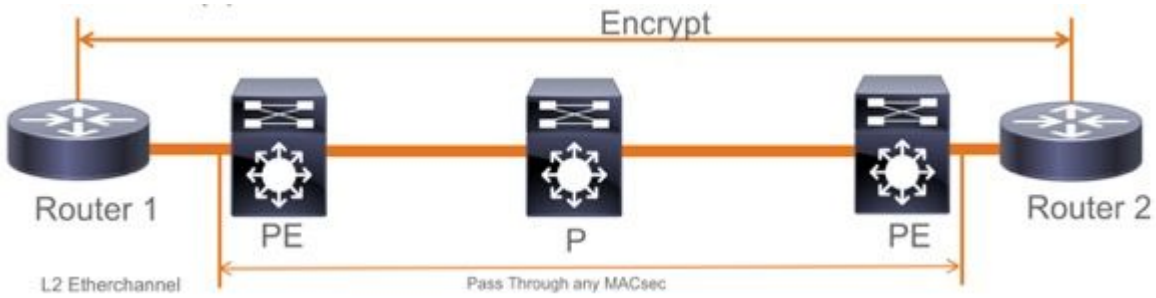




EoMPLS-/VPLS-Topologie

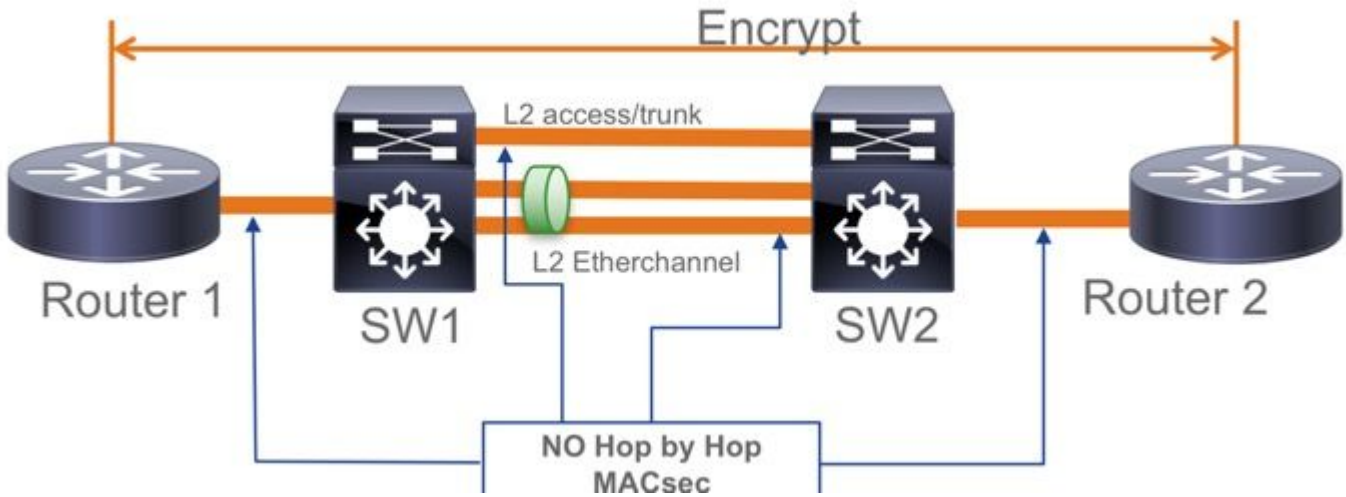
Unterstützte Plattformen, Catalyst 9300/9400, 9500/9500H als "**PE**"- oder "**P**"-Geräte

- VPLS
- EoMPLS
- Standardmäßig unterstützt (keine CLIs für Konfiguration zum Aktivieren/Deaktivieren)
- Start 16.10(1)

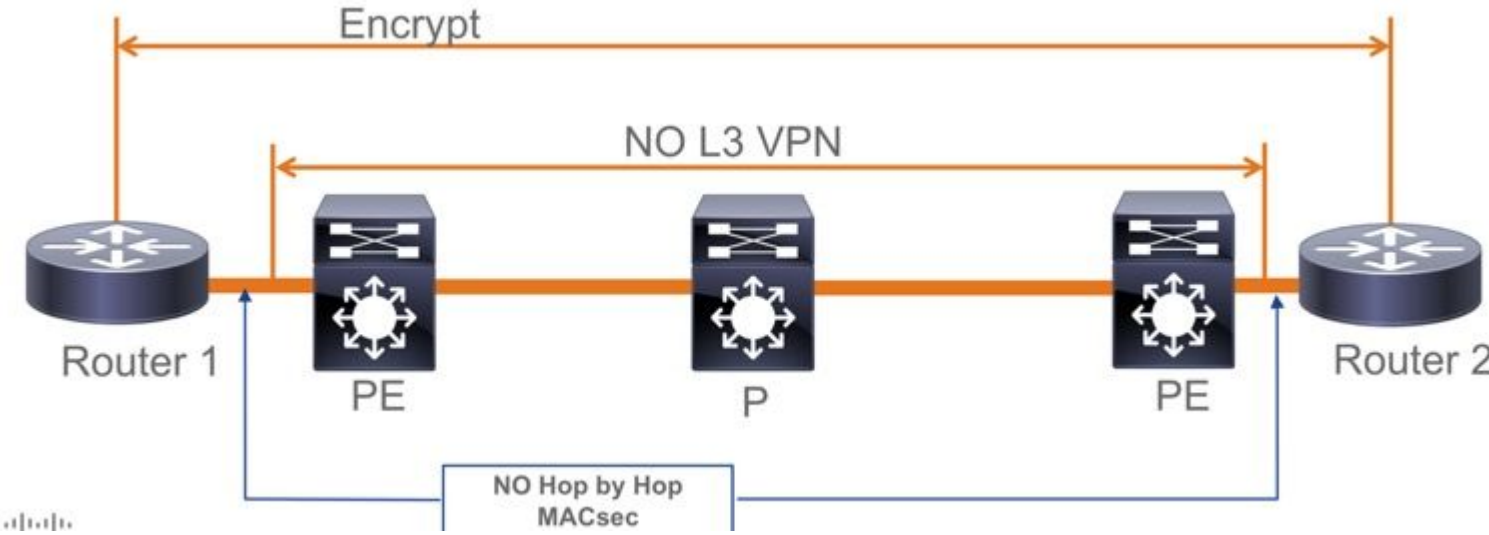


Einschränkungen

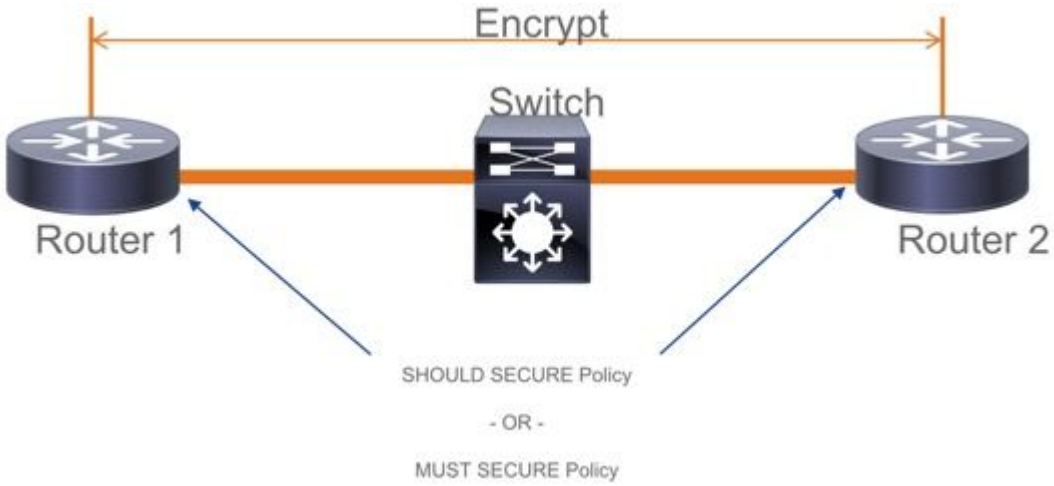
Doppelte Verschlüsselung wird nicht unterstützt. End-to-End-MACsec mit Clear-Tag erfordern, dass die Hop-by-Hop-Switches auf den direkt verbundenen L2-Verbindungen nicht aktiviert werden.



- ClearTag + EoMPLS mit nur Layer-2-Switches für mittlere Anforderungen; MACsec kann auf CE-PE-Verbindung nicht aktiviert werden
- ClearTag + L3VPN mit nicht unterstützten zwischengeschalteten Switches



- Im PSK-Modus wird "Sollte sicher sein" nicht unterstützt. Der Standardmodus ist "Must Secure".
- Muss die Sicherheitsrichtlinie nicht nur EAPoL verschlüsseln, um die MACsec-Einstellungen auszuhandeln

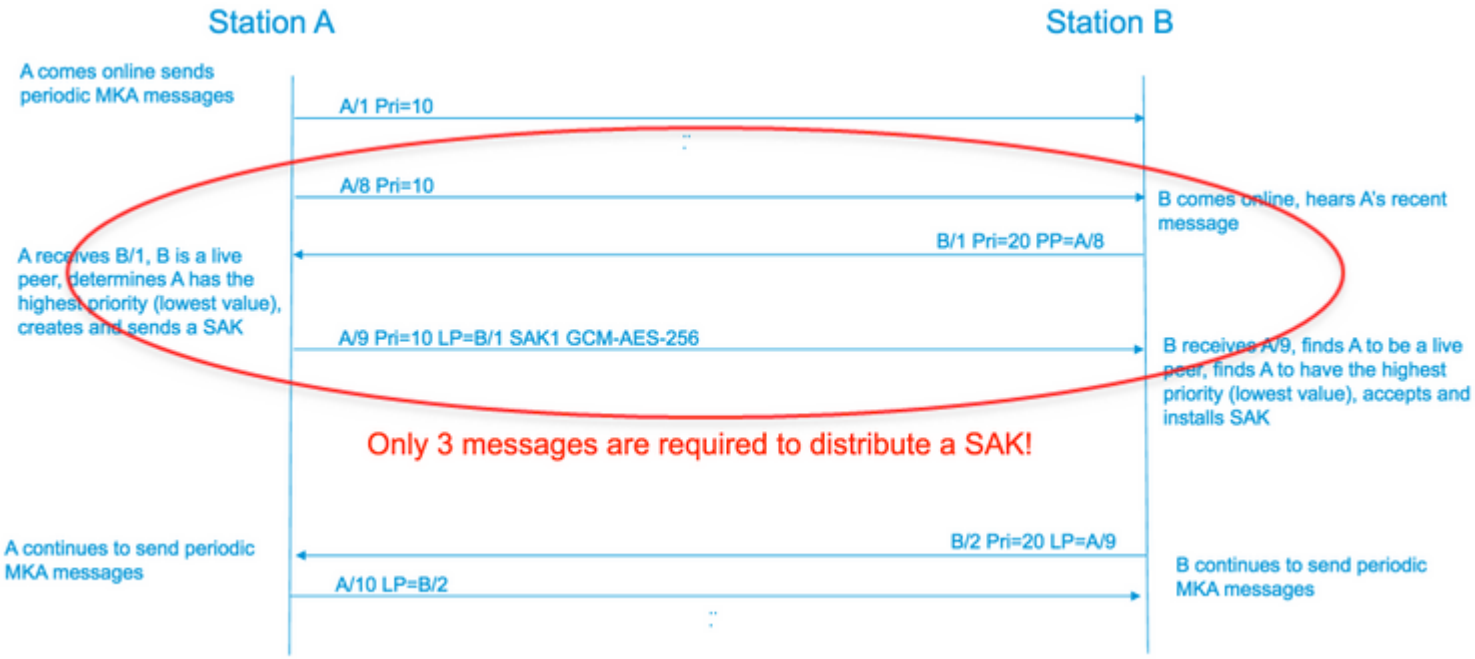


MACsec-Betriebsinformationen

Reihenfolge des Vorgangs

1. Wenn die Verbindung und beide Endgeräte aktiviert werden, tauschen sie MKA-Frames aus (**Ethertype = 0x888E,** identisch mit EAPOL mit Pakettyp wie MKA). Es handelt sich um ein Mehrpunkt-zu-Mehrpunkt-Verhandlungsprotokoll. Der Wert für den CAK-Schlüssel (normalerweise statisch, vorab freigegeben) und der Schlüsselname (CKN) müssen übereinstimmen, und die ICV muss für Peers gültig sein, damit diese erkannt und akzeptiert werden.
2. Das Gerät mit der niedrigsten Schlüsselserverpriorität (Standard = 0) wird als Schlüsselserver ausgewählt. Der Key-Server generiert das SAK und verteilt es über MKA-Nachrichten. Im Falle der Zeit gewinnt der höchste Wert von SCI (Secure Channel Identifier).
3. Anschließend werden alle MacSec gesicherten Frames mit der SAK (Symmetric cryptography) verschlüsselt. Es werden separate TX- und RX Secure Channels erstellt. Dasselbe Schlüssel-SAK wird jedoch sowohl für die Verschlüsselung als auch für die Entschlüsselung verwendet.

4. Wenn ein neues Gerät in einem Multi-Access-LAN erkannt wird (durch EAPOL-MKA-Nachrichten), generiert der Schlüsselserver einen neuen Schlüssel, der von allen Geräten verwendet werden kann. Der neue Schlüssel wird nach der Bestätigung durch alle Geräte verwendet (siehe Abschnitt 9.17.2 des IEEE-Standards 802.1X-2010).



MACsec-Pakete

Steuerrahmen (EAPOL-MKA)

- EAPOL-Ziel-MAC = 01:80:C2:00:00:03 für Multicast-Pakete an mehrere Ziele
- EAPOL-Ethertyp = 0x888E

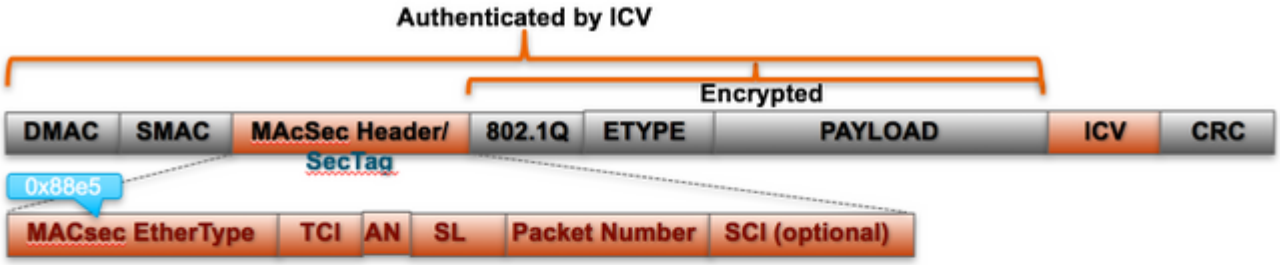
L2-Nutzlast im Control Frame-Format

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**

Datenrahmen

MACSec fügt zwei zusätzliche Tags in Datenframes mit maximalem Overhead von **32 Byte (**min. 16 Byte) ein.

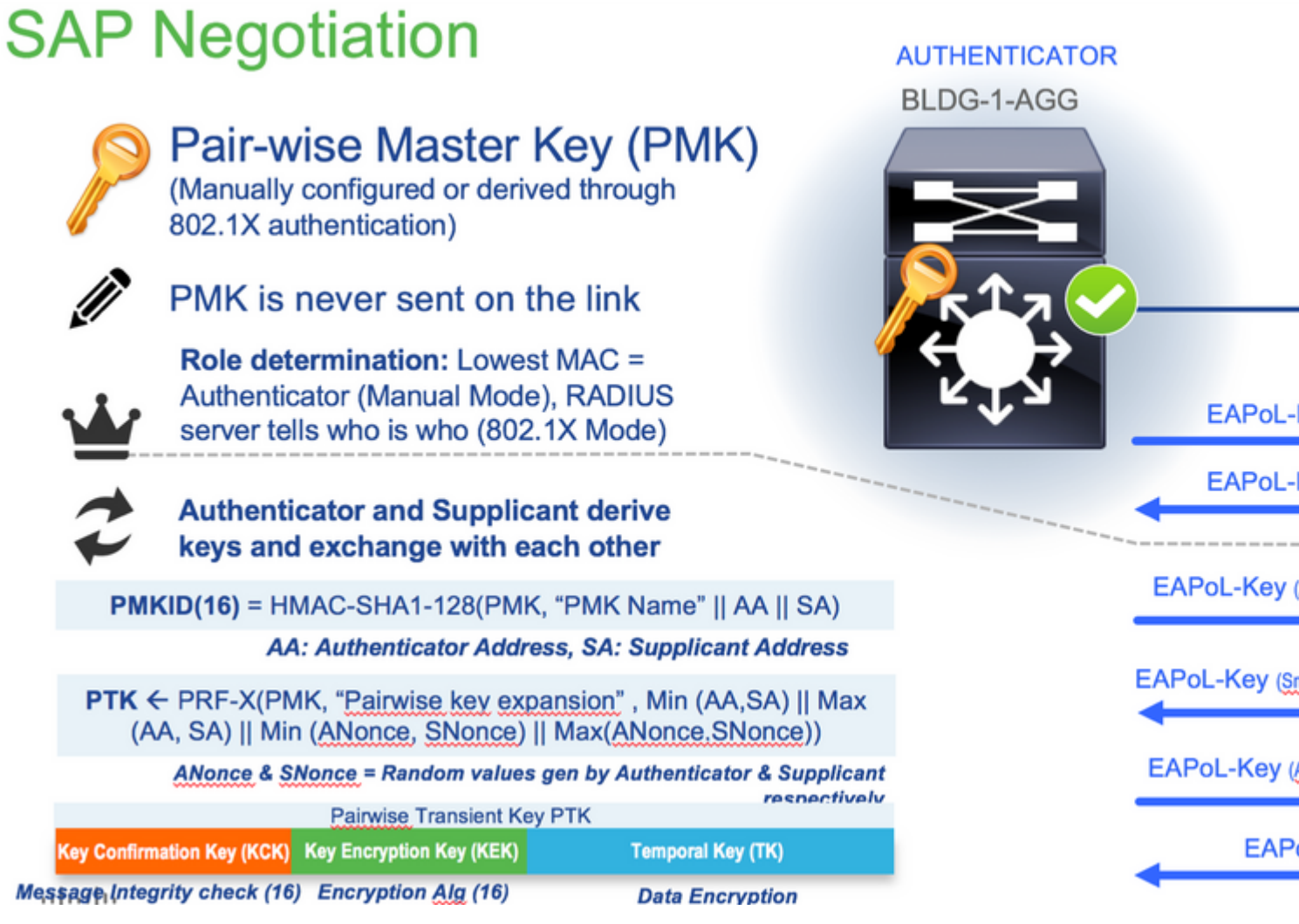
- **SecTag** = 8 bis 16 Byte (8 Byte SCI ist optional)
- **ICV** = 8 bis 16 Byte, basierend auf der Chiffrierfarbe (AES128/256)



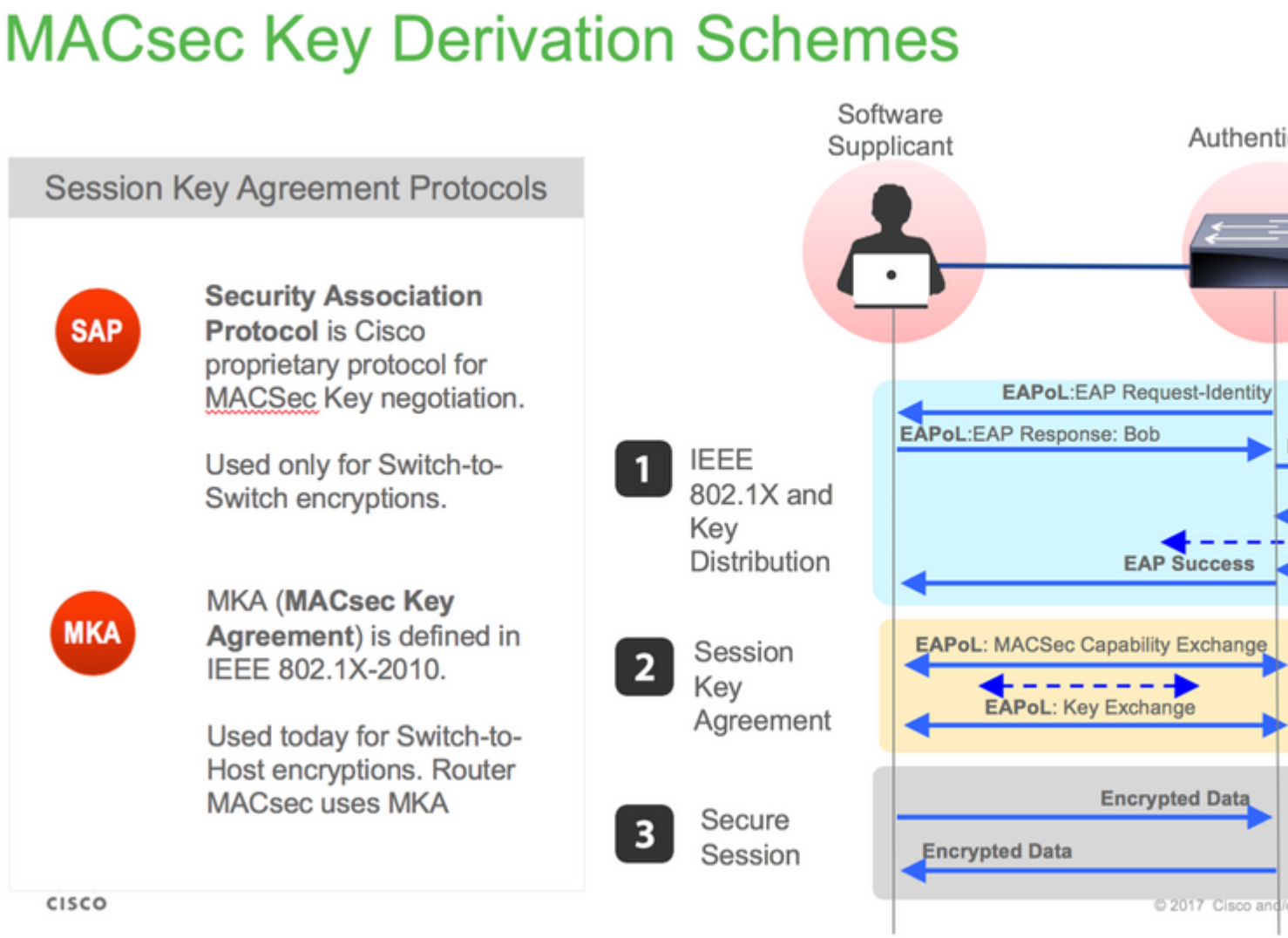
MACsec Tag Format

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)

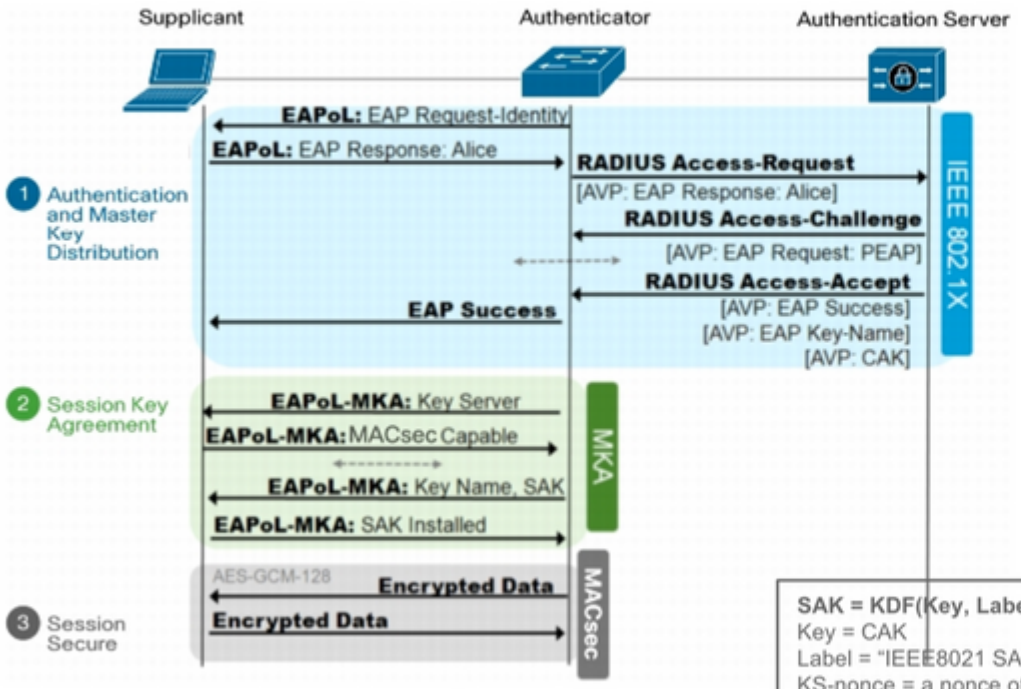
SAP-Verhandlung



Schlüsselaustausch



MKA Exchange



ICV = AES-CMAC(ICK, M, 128)
M = DA + SA + (MSDU – ICV)

A pairwise CAK (Connectivity Associatio
CAK = KDF(Key, Label, mac1 | ma

Key = MSK[0-15] for a 128 bit CAK, MSI
Label = "IEEE8021 EAP CAK"
mac1 = the lesser of the two source MA
mac2 = the greater of the two source MA
CAKlength = two octets representing an
256 bit CAK) with the most significant oc

The KEK(Key Encryption Key) is derive
KEK = KDF(Key, Label, Keyid, KEKLe

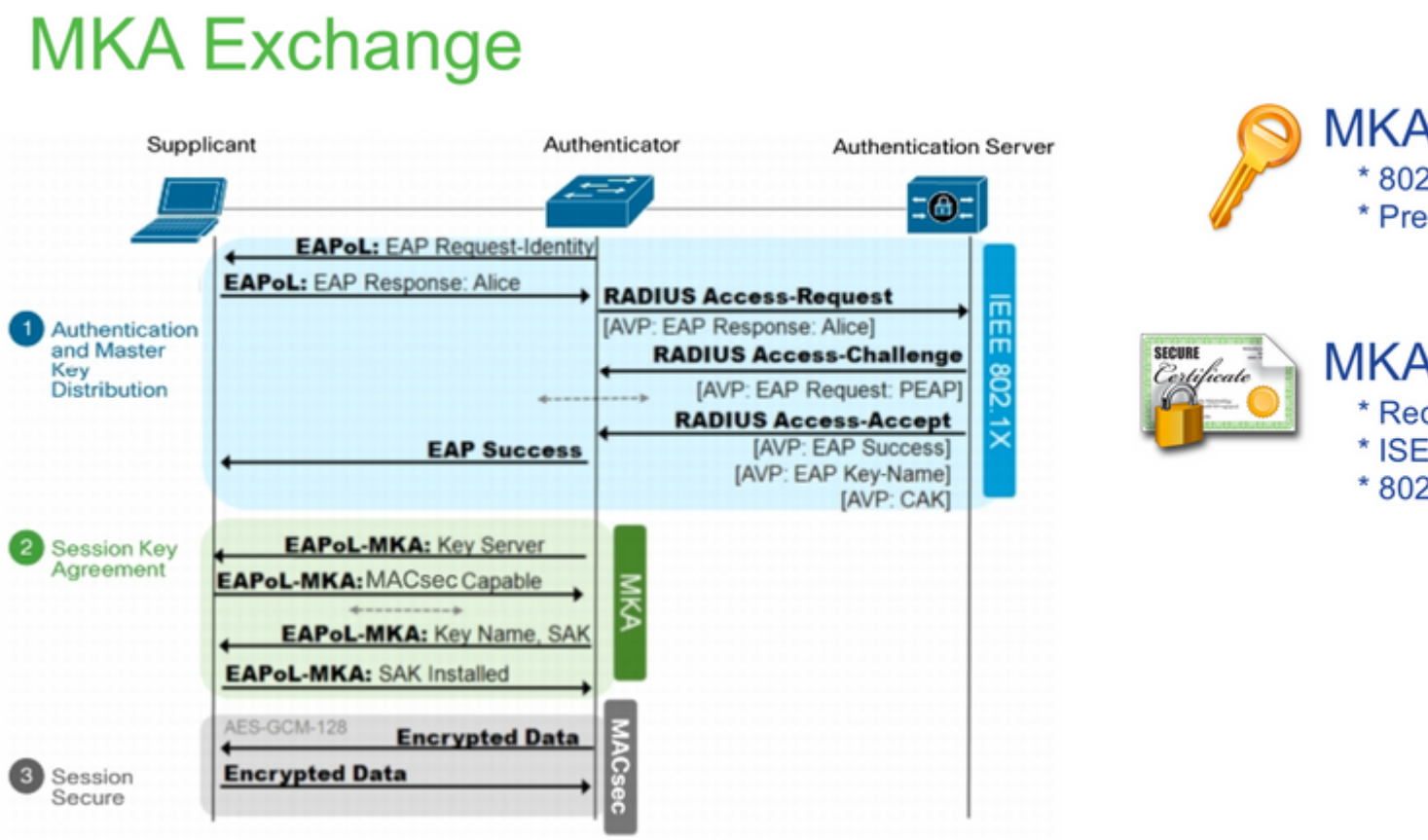
Key = CAK
Label = "IEEE8021 KEK"
Keyid = the first 16 octets of the CKN, w
KEKLength = two octets representing ar
256 bit KEK) with the most significant oc

The ICK (ICV Key) is derived from the C

ICK = KDF(Key, Label, Keyid, ICKLeng

Key = CAK
Label = "IEEE8021 ICK"
Keyid = the first 16 octets of the CKN, w
ICKLength = two octets representing an
256 bit ICK) with the most significant oct

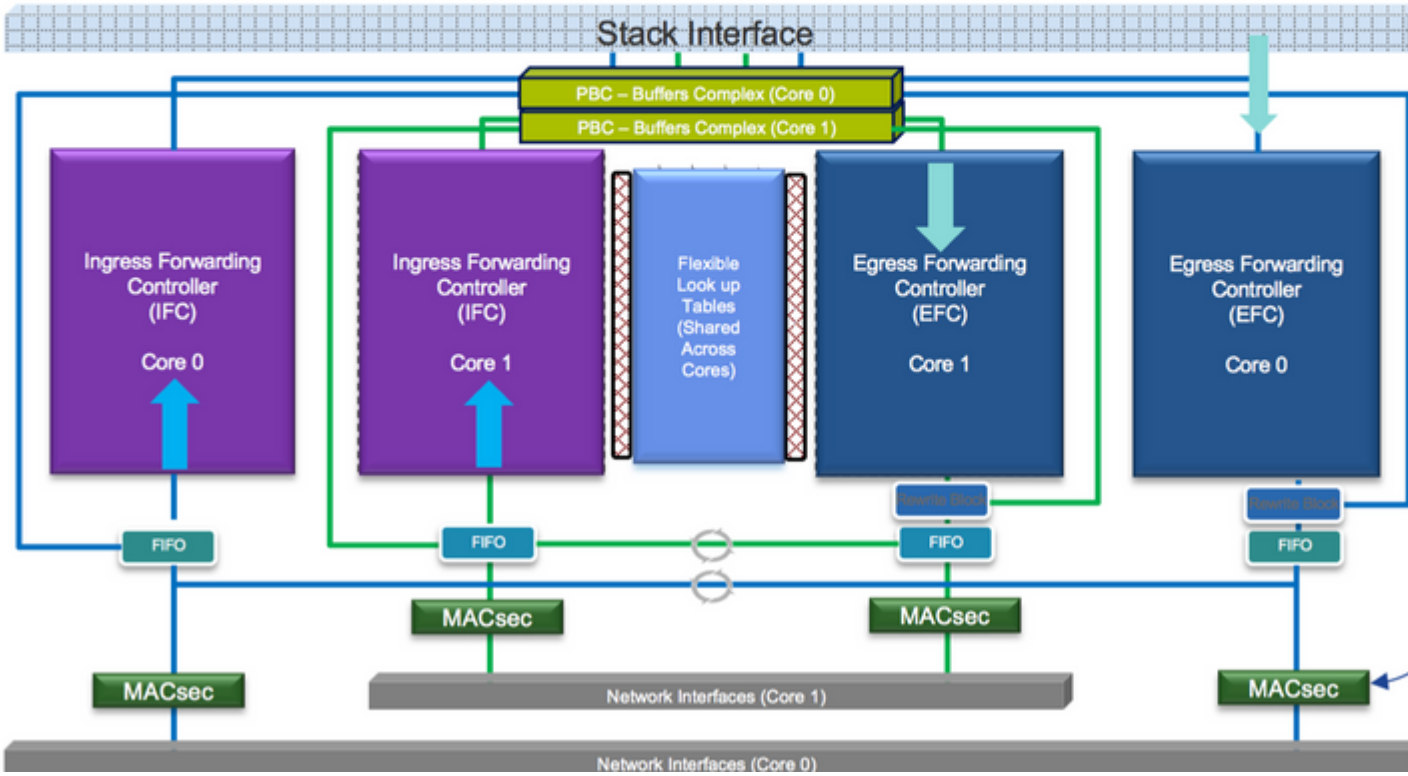
SAK = KDF(Key, Label, KS-nonce | MI-value list | KN, SAKleng
Key = CAK
Label = "IEEE8021 SAK"
KS-nonce = a nonce of the same size as the required SAK, obtain
MI-value list = a concatenation of MI values (in no particular order)
KN = four octets, the Key Number assigned by the Key Server as
SAKlength = two octets representing an integer value (128 for a 12
SAK) with the most significant octet first.



MACsec auf Plattform

Where is MACsec performed in Hardware?

Applicable for UADP 2.0/3.0/Mini ASIC



Produktkompatibilitätsmatrix

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 10/4

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

Zugehörige Informationen

Leitfaden zur Sicherheitskonfiguration, Cisco IOS XE Gibraltar 16.12.x (Catalyst 9300 Switches)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.