MACsec Enhanced

Authors	Hooman Bidgoli (Nokia), Hassen Clayton (Bell Canada), Nicklous Morris (Verizon), Ilijc Albanese (Telus), Nabbel Cocker (Redhat), Israel Meilik (Broadcom), Roii Werner (Broadcom)
Editor	Hooman Bidgoli
Date	
Document ID	

Change History

Version	Status	Date	Author	Owner	Reviewed by	Reviewed date	Approver	Approval date	Description of changes
0.1	Draft	26-08-2025	Hooman Bidgoli	Hooman Bidgoli	N/A	DD-MM-YYYY	N/A	DD-MM-YYYY	Init Draft

Contents

1 Terminology

Term	Definition
Upstream Router	From the datapath flow perspective, transmits packets to a downstream router
Downstream router	From the datapath flow perspective, receives packets from an upstream router
LSP	MPLS Label Switching Path
MKA	IEEE 802.1X - *MACsec Key Agreement protocol*
MPLS	RFC 3031 - *Multiprotocol Label Switching Architecture*
PE	Provider Edge router functions between one network service provider's area and the areas administered by other network providers
SAK	Secure Association Key used in MACsec
SCI	IEEE 802.1X - *Secure Channel Identifier*
Segment	Can represent any instruction, topological, or service-based entity within a network
SID	RFC 8402 - *Segment Identifier*
SR	RFC 8402 - *Segment Routing*
SRv6	RFC 8986 - *Segment Routing over IPv6 networks*
SLA	The Service Level Agreement is a formal contract between a service provider and a customer that defines the specific services to be delivered, the expected performance standards (for example, uptime and response times), and the responsibilities of each party.

2 Introduction

The IEEE 802.1AE 2006 MACsec standard specifies how a network can be secured transparently to peer protocol entities that communicate using the MAC service provided by IEEE 802 LANs. This applies to all or part of a network. MACsec provides high-performance, low-latency, end-to-end encryption at the MAC layer. In implementations, MACsec can leave the VLAN tags into clear

and without integrity check, thereby allowing VLAN manipulation and end-to-end packet forwarding within the VLAN switched network. MACsec requires both ends of the connection to have a MACsec-capable device, such as a switch, router, or a network interface card (NIC). These devices use a key agreement protocol, such as MKA, to establish and manage the encryption keys known as SAKs.

However, in a Layer 3 network that is interconnected via a set of routers, the MACsec encryption must terminate at the point-to-point interface of each router, to allow the routers to examine the Layer 2.5 MPLS or Layer 3 IP header and make appropriate routing decision. MACsec can be enabled on the egress interface of the upstream router, to encrypt the packet to the downstream router, and so on, until the packet reaches its destination in the Layer 2.5 (MPLS) network or Layer 3 (IP) network. In this scenario, the following considerations apply:

- Every router must be MACsec capable.
- The encryption is not end-to-end for the MPLS or IP flows. At each hop the packet flow is decrypted to allow the appropriate routing (Layer 3) or switching (Layer 2.5) decision.

These two points create a challenge for MACsec in an IP/MPLS network. The cost of deploying MACsec in an IP/MPLS network is high, as are the security risks because end-to-end encryption is not possible. One example of a security concern is the possible mirroring of secure flows to an unsecure port, while the flows are unencrypted on the router and before they are encrypted again on egress.

This white paper examines a new proposal to address these challenges, using the MACsec encryption engine and the IEEE802.1AE standard to encrypt the payload of the MPLS and IP packets, while leaving the Layer 2.5 MPLS label stack or the Layer 3 IP header in clear and without integrity protection. Using this approach, the transit routers can forward the packets without decrypting them, by examining the corresponding OSI layer header. This enables the IP and MPLS packets to remain encrypted end-to-end from the originating router to the terminating router.

3 Background motivation

Imagine having a single standard and implementation for line-rate and low-latency encryption, with key distribution for multiple OSI layers. Operators would only need to familiarize themselves with a single encryption solution, capable of matching any flow at any OSI layer and could identify and secure a flow in brownfield or greenfield networks with a simple configuration with proven MACsec encryption engine.

MPLS is the most dominant transport in vertical segments, including critical infrastructure and service providers. In addition to MPLS, SRv6 is gaining momentum in the service-provider segment. Both these technologies provide highly resilient transport with traffic engineering.

Security and encryption are becoming a more integrated part of the transport because of government enforcements and application standards, for example, 3GPP Control Plane (CP). At the same time, integrating security solutions into existing MPLS and IP/SRv6 networks is a significant challenge for most operators. The operators prefer to design their networks based on their service level agreement (SLA) requirements and transport of choice, and to seamlessly enable and integrate security using end-to-end encryption solutions without re-engineering their networks. Currently however, there is no viable solution for end-to-end MPLS or SRv6 encryption. The MACsec encryption engine provides line-rate and low-latency encryption, which is attractive for maintaining existing network SLAs. However, as previously described, MACsec itself is only a hop-by-hop encryption solution for Layer 2.5 MPLS and Layer 3 IP networks.

As chip technology evolves, the MACsec engines are now baked into IP/MPLS-capable chips, making it feasible, with minor modifications to the IEEE 802.1AE standard, to use the MACsec encryption engines to enable seamless encryption at multiple layers of OSI, including the MPLS and IP layer. For key distribution, MKA can be carried over IP/UDP to secure key delivery over an IP/MPLS network. This solution is attractive to operators because they can enable MPLS or even IP/SRv6 encryption on their existing brownfield networks by software upgrade and select any existing flow within their networks for encryption. They simply identify the flow and create a connectivity association for that flow. Using a simple configuration, operators can enable encryption and security in their existing IP/MPLS networks, without affecting their existing SLAs, and more importantly, without redesigning their entire networks.

This white paper examines the proposed changes required to allow IEEE 802.1AE to be used to encrypt MPLS and IP flows end-to-end, and MKA to be used as a reliable key distribution in an IP/MPLS network.

4 High-level design proposal

As more chips implement the MACsec engine within the same ASIC that can forward IP/MPLS packets, it is possible for the chips to match specific IP or MPLS flows for encryption by the MACsec encryption engine. The chips are capable of dynamically programming the encryption and integrity offset and positioning the security tag (SecTAG) in certain byte offsets of the packets. This makes it possible to encrypt the payload of the IP or MPLS packet after the SecTAG

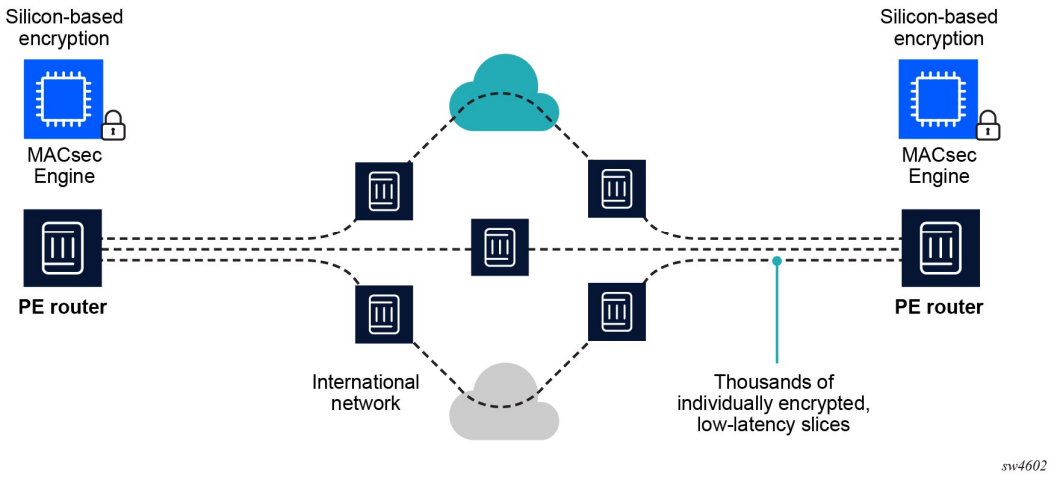
and run integrity protection on the SecTAG and payload of the IP or MPLS packet, while leaving the IP header or the MPLS label stack in clear and without integrity protection.

NIST-approved MACsec algorithms provide the encryption and authentication function. These algorithms include the following:

- IEEE802.1AE 2006 GCM-AES-128
- IEEE802.1AEbn 2011 GCM-AES-256
- IEEE802.1AEbw 2013 GCM-AES-XPN-128/256

The following figure shows the encryption process.

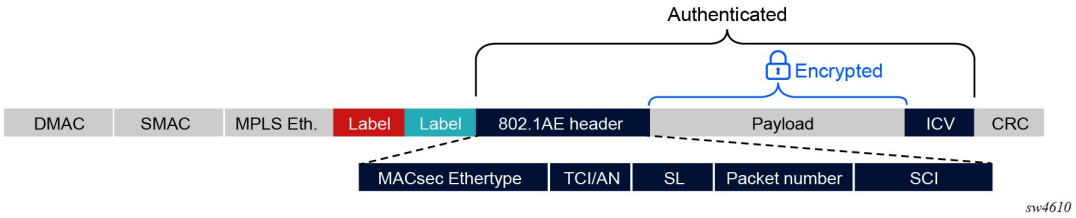
Figure 1 IP/MPLS Flows encrypted end-to-end with MACsec encryption engine



By leaving the IP header or MPLS label stack in clear, the transit or LSR routers can switch the packet based on the IP header or the label stack. For MPLS specifically, the LSR router can SWAP

or POP the label as necessary to forward the packet from one PE router to another. The following figure shows the MPLS packet secured by the MACsec encryption engine.

Figure 2 MPLS packet secured by MACsec encryption engine



The IP or MPLS protocols are implemented based on the existing IETF documents, such as RFC8402 Segment Routing. As such, these IP or MPLS protocol are capable of automatically calculating the location of the payload and the encryption and integrity protection offset. In short, these IP or MPLS protocols are fully aware of the IP header or label stack size. As an example for MPLS, the protocol is aware of the number of labels in the label stack of the packet that must be pushed for the packet to successfully traverse the network and reach its destination. These protocols program the ASIC's IP or MPLS forwarding tables with appropriate label-stack or IP-forwarding information. This makes it possible for these protocols to "DYNAMICALLY" calculate the encryption and integrity offset, and program the MACsec engine with it, as well as the byte offset of where to position the SecTAG. Unlike the previous "SELECTABLE" confidentiality offset, where the user could configure to (0, 30, 50) of the initial octets of the MSDU, in the new proposal, the protocol layer does all the calculation and provides it to the MACsec engine "DYNAMICALLY". This makes the "SELECTABLE" offsets unnecessary.

In summary, the new chip capability and protocol knowledge for encrypting an IP or MPLS payload and positioning the SecTAG at the correct offset for each OSI layer is a dynamic and intuitive process. The integration of the IP/MPLS forwarding logic and MACsec encryption engine in a single chip makes it possible for the new implementations to use the MACsec engine to secure MPLS, IP, and even SRv6 flows end-to-end through a Layer 2.5 or Layer 3 network. However, using IEEE802.1AE for IP/MPLS encryption breaks some standard rules, and these rules must therefore be modified to fully allow encryption of IP/MPLS flows with this technology. The following sections examine the requested changes to existing MACsec standard in more detail.

5 Proposed changes to IEEE802.1AE

The topics in this section are proposed changes to the current IEEE 802.1AE standard.

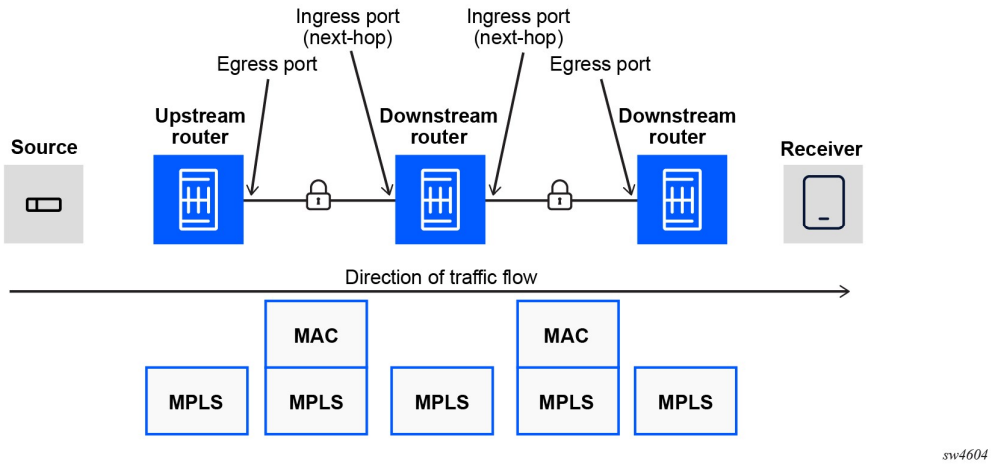
Most of these proposed changes have been verified in existing networks with existing implementations.

5.1 Leaving MAC addresses in clear and without integrity protection

In IEEE802.1AE 2006 section 8, figure 8-1, currently the mac addresses are mandated to be part of the integrity protection.

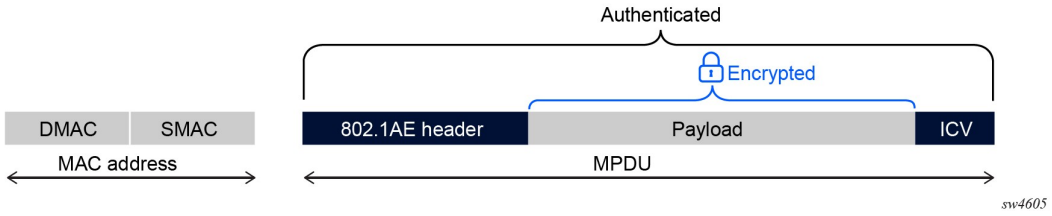
In a routing domain, the MAC address forwards a Layer 2.5 MPLS or Layer 3 IP packet from the egress port of the upstream router to its ingress (next-hop) port on the downstream router.

Figure 3 End-to-end MPLS (Layer 2.5) forwarding



The downstream router strips the MAC address and makes a Layer 2.5 MPLS switching decision or a Layer 3 IP routing decision to find the egress port of the packet on the router. At the egress port of this router, a new source and destination MAC address are appended to the packet, based on the source MAC address of the egress port and the destination MAC address of the next-hop port on the downstream router.

Figure 4 Proposed new figure for IEEE 802.1AE



For an end-to-end MPLS or IP flow encryption using the MACsec encryption engine, the MAC address must not be part of the integrity protection. This makes it feasible for the routers to strip the MAC address and insert a new MAC address at the switch or route the packet to the destination IP or MPLS PE.

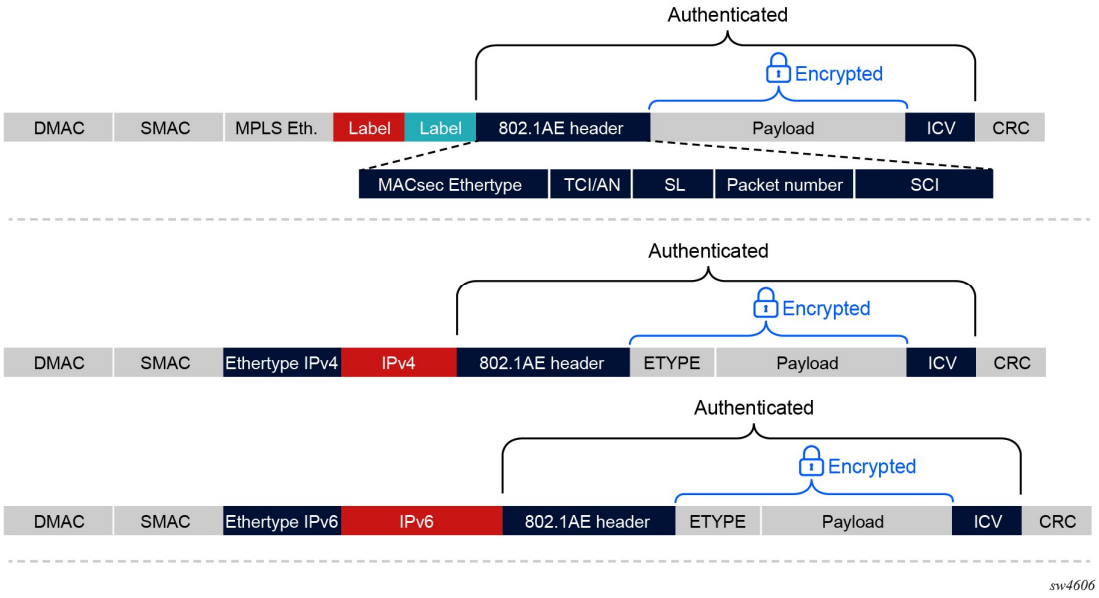
5.2 Leaving MPLS Label stack or IP header in clear without integrity protection

In IEEE 802.1AE 2006 section 8 figure 8-1 and its footnote points out that the L2 MAC header (MAC Addresses) and the MPDU are seprate componants and there can be additonal octets injected between them.

This explanation with the new diagram proposed in 4 High-level design proposal, may be sufficient to leave the MPLS label stack or the IP header in clear between the MAC addresses and MPDU field, where the secure data is the payload of the MPLS or IP packet.

That said, the following figure provides examples of the new encapsulations proposed for IP and MPLS in this white paper. As explained in 3 Background motivation, the MPLS or IP protocols dynamically calculate the number of MPLS labels or the size of the IP header that must be in clear and without integrity protection, and program the offset of encryption and integrity protection into the MACsec encryption engine. As such, the new description should not use “SELECTABLE” offset. Instead, it must indicate that the appropriate IP/MPLS protocol works hand-in-hand with the MACsec encryption engine to calculate and program the necessary encryption and integrity protection offset of the frame and provide the position of the SecTAG within the frame stack.

Figure 5 New encapsulation proposal for IP/MPLS

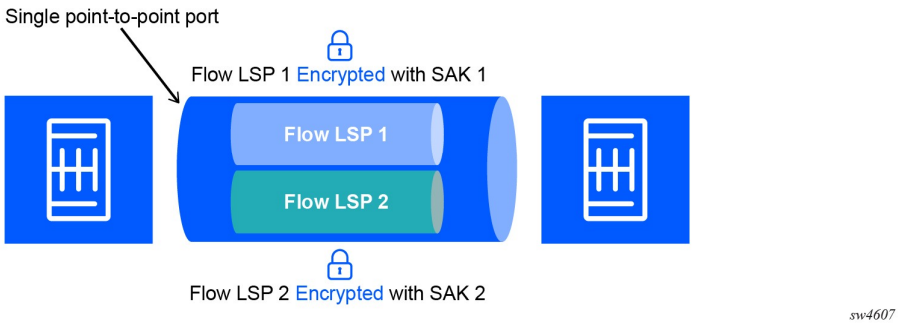


5.3 Changes to the SecTAG

IEEE 802.1AE 2006, section 9.3 mentions that the encoding of the SCI field in the SecTAG is optional.

IEEE 802.1AE 2006, section 9.9 mentions that an explicitly encoded SCI field in the SecTAG is not required for point-to-point links.

Figure 6 Multiple LSPs using a single MACsec port for encryption and decryption



For MPLS or IP encryption, even in the case of point-to-point connection, the SCI field is mandatory. Many MPLS or IP flows can be encrypted on the same egress port of the MACsec encryption engine or can terminate on the same ingress port of the decrypting downstream router. Each flow must be uniquely identified with a SCI to its corresponding Security Association (SA); see 5.4 Changes to the SCI for more information.

5.4 Changes to the SCI

IEEE 802.1AE 2006 section 9.9 explains the encoding of the SecTAG, where Octets 9 to 14 is a system identifier, as an example a unique MAC address and Octets 15 to 16 is a port identifier, as an example the vlan id or the port id.

For MPLS or IP encryption, the SCI must be redefined to identify an LSP or an IP flow uniquely. For an MPLS LSP, the current implementation uses the LSP ID as per RFC 3209, and an encryption SID to build a unique SCI.

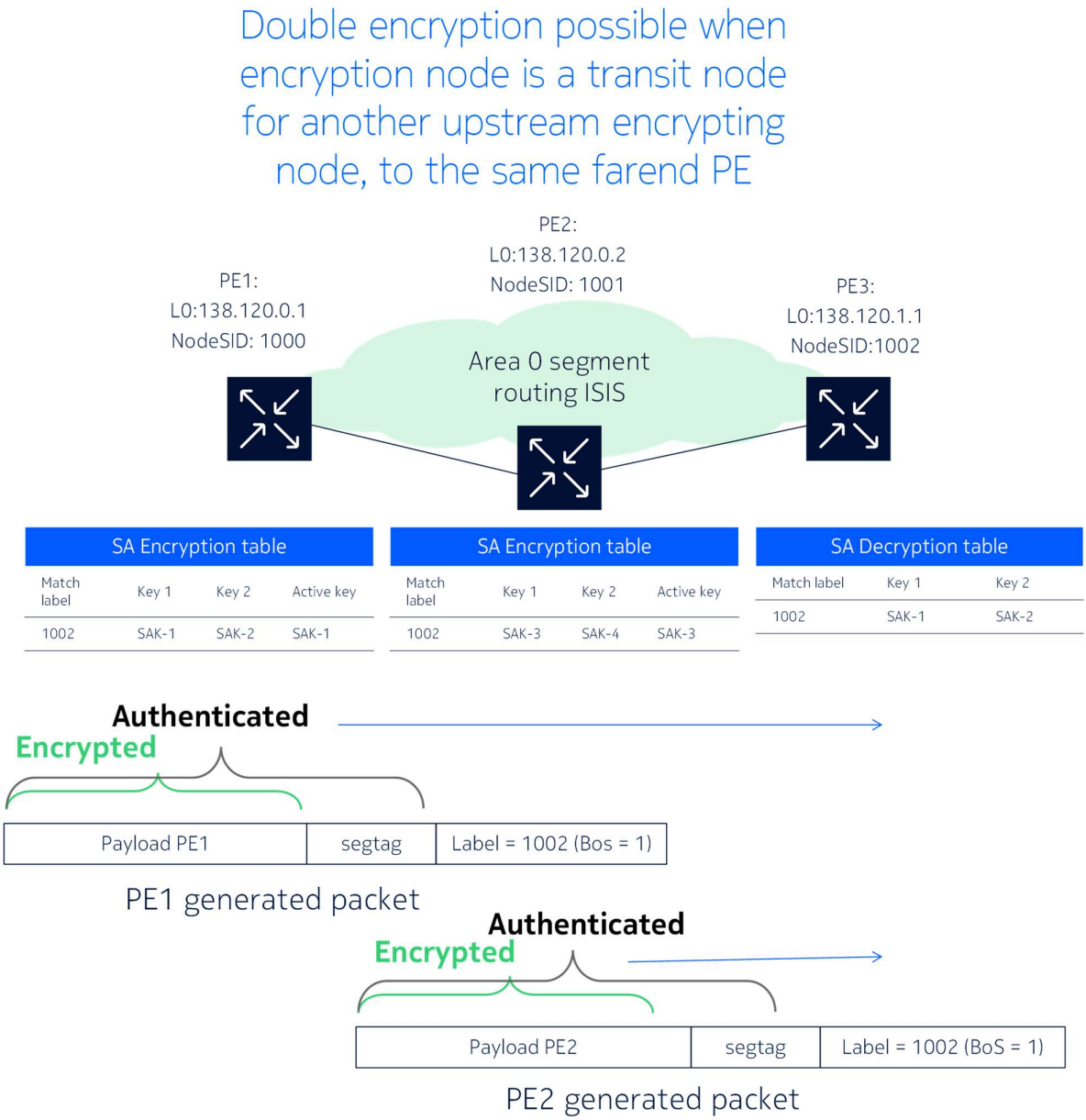
- The LSP ID is a 16-bit identifier that identifies the LSP uniquely on the upstream (encrypting) PE
- A SID is defined in RFC 8402. A segment is often referred to by its segment ID (SID). A segment can represent any instruction, topological or service based. An encryption SID identifies an encryption entity uniquely within the network. Depending on the implementation it can be assigned “uniquely per router or even per encryption engine or encryption flow. The encryption SID is not yet defined in IETF but it will be soon by introducing a new draft.

The combination of these two fields can uniquely identify an encryption flow within the network and the associated SecY.

The encoding of SCI for MPLS and IP flows can be farther discussed in detail in future.

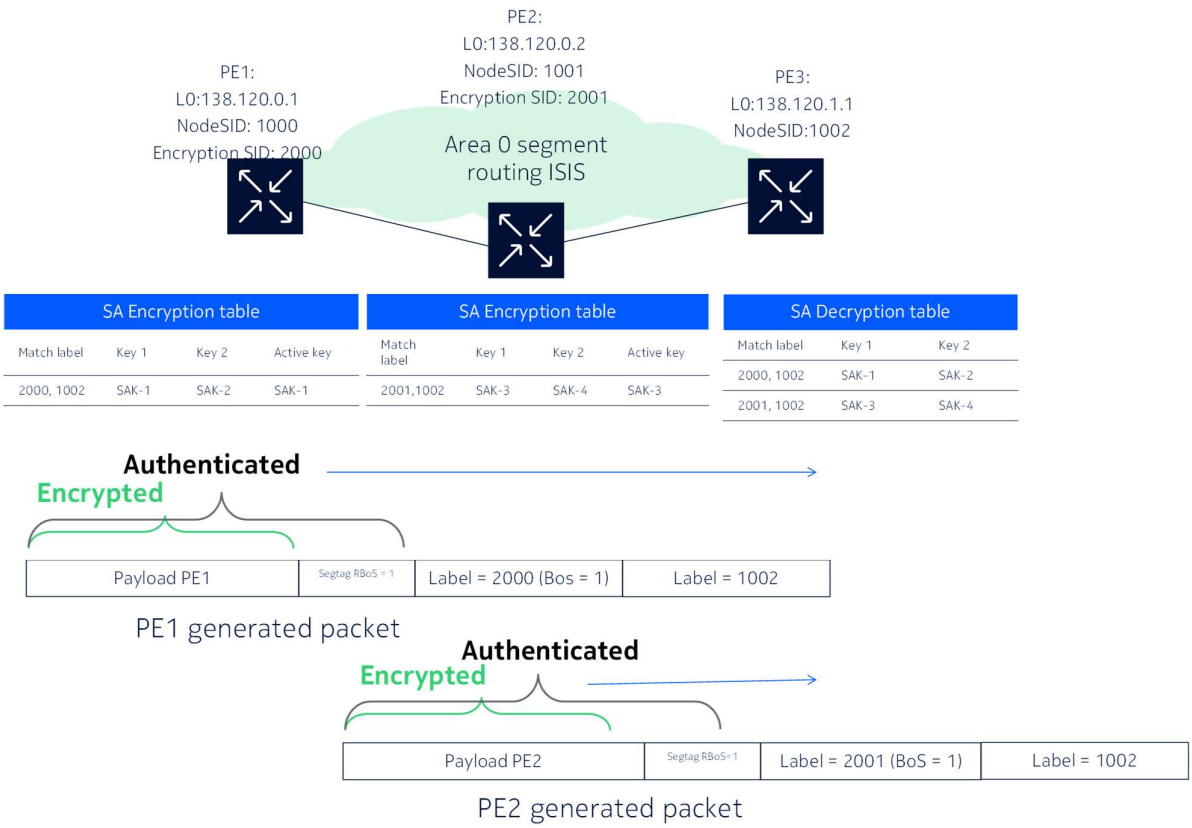
5.5 Storing and Restoration of MPLS or IP information in SecTAG

Figure 7 Double Encryption scenario



In some cases, like Segment Routing (SR) the same label stack can be used from two or more encrypting PEs to the same destination PE. Since the label stacks are identical from upstream routers compared to the locally generated label stacks, double encryption can occur as per figure 7.

Figure 8 Encryption SID and SecTAG Restore Bottom of Stack Bit



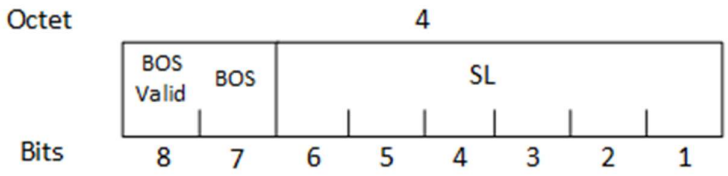
A network unique encryption SID is inserted by an encrypting router to uniquely identify the source encrypting router. This identification is important to ensure double encryption does not occur on the network. In this case the encryption SID is used to create a unique label stack for the encrypting router throughout the network, so if the packet from this encrypting router traverses another downstream encrypting router which is also encrypting packets to the same destination PE, double encryption does not occur.

If a MPLS flow is selected for encryption, the encryption SID is usually injected at the bottom of the stack with the bottom of the stack (BoS) bit set. Where previously the BoS label was a transport label with BoS bit set, now the encryption SID has the BoS bit set and the transport label has BoS cleared.

The decrypting router removes the encryption SID after decrypting the packet and must restore the BoS bit on the transport label. To do so the encrypting router must store the information that the BoS bit must be restored on the transport label, in the SecTAG. In the current

implementation two bits from the Short Length (SL) is stolen (i.e. the most significant bit in SL) to signal if the BoS bit must be restored by the decrypting router or not. This might be acceptable.

Figure 9 BOS stored information in SL



IEEE 802.1AE-2006 section 9.7 has mentioned that Bits 7 and 8 of octet 4 shall be zero. For layer 2 MACsec encryption. For layer 2.5 MPLS encryption, we purpose these unused bits for the BOS flag. When bit(8)=1, then bit(7) conveys the BOS value from the original ES Label.

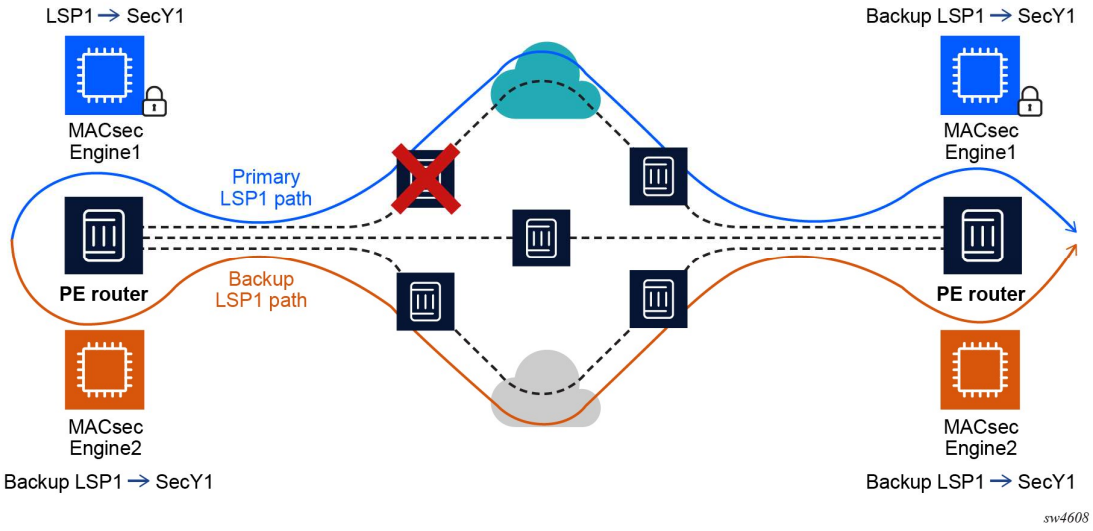
When bit(8)=0, then no BOS information is conveyed and bit(7) is assumed to be 0.

5.6 Changes to the packet number in the SecTAG

NOTE: The authors understand that changes to the field size of packet number might be difficult to achieve. This section is just a proposal and there might way other ways to solve this issue.

The following figure shows an example of synchronizing the packet number (PN) over multiple MACsec encryption engines, for example, when using backup paths for resiliency.

Figure 10 MPLS flow encrypted by MACsec Engine 1 being protected by Engine 2



Another complex issue with MPLS or IP flow encryption using the MACsec encryption engine is synchronizing the PN over multiple MACsec encryption engines.

In Layer 2.5 MPLS and Layer 3 IP, resiliency and fast recovery is an important concept. IP and MPLS flows can be protected by backup paths through a network. For example, if an IP or MPLS flow has a primary next hop through port 1 of a router and port 1 is encrypted via the MACsec engine 1 with appropriate SecY1, this flow can have a protection next hop through port 5 of the router, where port 5 is encrypted via MACsec engine 2 with the same SecY1.

In another example, if technologies such as Equal Cost Multi Path (ECMP) or Link Aggregation Group (LAG) are used on the encrypting router, the MPLS or IP flow can be sprayed over multiple ports or interfaces, each with its own encryption engine. All encryption engines on all ports must encrypt the same flow with the same SecY. This means that the same SecY must be downloaded on multiple encryption engines.

As these examples show, to ensure all packets of a single IP or MPLS flow are encrypted with a unique SecY and unique IV PN, there must be a mechanism to synchronize the PNs between multiple encryption engines. This dynamic recovery and load-balancing behaviour of IP/MPLS is not the same as the MAC layer. At the MAC layer, the MACsec encrypts a Layer 2 flow on a single port and a single encryption engine. As an example for LAG, each LAG member is encrypted with its own SecY and MACsec session.

Consider a 400 GB line rate IP or MPLS flow with fast recovery or load balancing, where SecY1 is used for encryption of the flow.

1. On the egress, SecY1 must be downloaded to all ports and encryption engines that could be used to encrypt the flow in case of fast recovery or load balancing.
2. On ingress, the IP and MPLS forwarding instruction is usually downloaded to all ports. This is because IP and MPLS packet forwarding throughout a Layer 3 network is dynamic and these flows can ingress to the destination router on any port. Hence the SecY1 needs to be available on all encryption engines for these ports.
3. On the egress, the PN must be synced between the encryption engines used for encrypting that specific flow, to ensure the same SAK and PN are not used on two different encryption engines to encrypt the outgoing packets.
4. At 400 GB GCM-AES-XPN should be used. In XPN, the most significant 32-bit of the PN is locally tracked and the least significant 32-bit of the PN is transmitted in the SecTAG PN field. The most significant 32-bit of the PN should be tracked and synchronized on all ports and encryption engines using SecY1 for encrypting a single flow with a single SAK.

To solve these issues, there must be a PN synchronization method between the encryption engines. Currently this mechanism is in software, which at 400 GB line rate has approximately five seconds or so to synchronize the least significant 32-bit PN between encryption engines. This is a complicated and cumbersome process for the short period of time.

Given the dynamic routing and load balancing capabilities of IP/MPLS, and the high encryption rates available today (for example, 400 GB or even 800 GB) it is extremely important to introduce a 64-bit (8 octets) packet number in the SecTAG. By using a 64-bit PN in the SecTAG, it is possible to assign a range of PNs to each encryption engine used for encrypting the same IP/MPLS flow. The following is an example.

Section	encryption id	PN
Bit:	|63|62|61|60 |	|03|102|01|101|
Port 1	| 0 | 0| 0| 1 |	PN for port 1
Port 2	| 0| 0| 1| 0 |	PN for port 2
. . .		

In the example, encryption engine 1 is presented by bit 60 in the 64-bit PN, encryption engine 2 is presented by bit 61 in the 64-bit PN, and so on. Using this method, each encryption engine has its own packet number identifier via the encryption ID on the most significant bits (in the example, using the four most significant bits for encryption ID) and 60 bits of packet number for that port PN. This provides a large PN range that is unique for each encryption engine, because each port in the LAG bundle or ECMP group has its own PN pool. If the IP or MPLS flow packets are sprayed over the two ports and the encryption engines, there is no need for PN syncing on each port because each port PN is unique based on the encryption ID.

This is one example; other implementations can also take advantage of the 64-bits packet number.

For backward compatibility the chip can function in 64-bits PN or XPN mode.

5.7 Example MPLS packet encrypted via the MACsec engine.

The following description and figure provide an example of an MPLS packet encrypted with the MACsec encryption engine.

1. The MAC addresses are in clear without integrity protection.
2. The label stack is in clear without integrity protection.
3. The MPLS protocol, in this case RFC 8402 *Segment Routing,* gives the encryption offset and the integrity protection offset to the MACsec encryption engine.
4. The MPLS protocol also calculates the SecTAG offset.
5. The SCI in the SecTAG is LSP-ID + Encryption SID.
6. The encryption SID is a 20 bit label and uniquely identifies the source of encryption with in the network. In below packet capture the encryption SID is 3rd in the label stack with a value of 51202 = HEX 0x0C802.
7. It can be observed in the SecTAG SCI field that the value is 0x0C8020004, where 0x0C802 is the encryption SID and the LSP-ID is the value 0004.
8. The combination of encryption SID and LSP-ID identifies the encrypting source and the unique LSP within that encrypting source.

Figure 11 MPLS Segment Routing packet encrypted with MACsec engine

```
        67 33.617…  1.0.12.1                 1.0.12.2            EAPOL-…    128 Key Server
        68 33.940…  1.0.12.2                 1.0.12.1            EAPOL-…    208 Live Peer List, MACsec SAK Use, XPN, ICV Indicator
        69 34.550…  Nokia_74:9c:1c           Nokia_03:90:da      MACSEC     150 MACsec frame
<
> Frame 69: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface ens16, id 0
> Ethernet II, Src: Nokia_74:9c:1c (9c:e0:41:74:9c:1c), Dst: Nokia_03:90:da (a8:24:b8:03:90:da)
> MultiProtocol Label Switching Header, Label: 60221, Exp: 7, S: 0, TTL: 255
> MultiProtocol Label Switching Header, Label: 51202, Exp: 7, S: 0, TTL: 255
> MultiProtocol Label Switching Header, Label: 7 (Entropy Label Indicator (ELI)), Exp: 7, S: 0, TTL: 255
> MultiProtocol Label Switching Header, Label: 922644, Exp: 7, S: 1, TTL: 0
> 802.1AE Security tag
> Data (88 bytes)
```

```
0010  de ff 0c 80 2e ff 00 00  7e ff e1 41 4f 00 88 e5   ····.··· ~··AO···
0020  2c 00 00 00 00 01 00 00  00 00 c8 02 00 04 98 41   ,····· ·····A
0030  30 7e 96 36 01 01 c1 7f  07 72 aa a9 69 00 1d 93   0~·6···· ·r··i···
0040  f8 bb 6e ee 48 ae f4 dd  d6 ac 1a 11 9b ff 17 55   ··n·H··· ·····U
0050  54 13 5a ab 8a d4 d1 6e  29 93 b6 ac 96 5b 46 f5   T·Z····n )····[F·
0060  db 17 0d 2b 5b c0 66 48  4e 32 7e 01 c9 60 f1 31   ···+[·fH N2~··`·1
0070  ba 59 22 13 46 85 ae c7  c2 28 99 1b 7f 98 bf 74   ·Y"·F··· ·(·····t
0080  6d 16 c6 b8 ae e0 a6 cb  fa d7 08 29 64 2f 6c b2   m····· ···)d/l·
0090  d8 90 28 cc a6 88                                  ··(···
```

6 Proposed changes to IEEE802.1X

To use the MACsec engine to encrypt IP and MPLS flows from one PE to another PE that is multiple hops away, it is necessary to make minor modifications to the MKA protocol. The following sections describe minor modifications to the IEEE802.1X standard that are required for the purpose of transporting MKA over a Layer 3 IP network to distribute SAKs between PEs.

6.1 MKA encapsulation

IEEE802.1X-2010, section 6.3.2, mentions MKA is conveyed by EAPOL PDUs and it is distinguished from PACP by EAPOL Packet Type which is defined in IEEE802.1x-2010 section 11.3.2, Table 11-3.

EAPOL packets are Layer 2 PDUs that are not routable in a Layer 3 network. To make MKA packets routable in a Layer 3 network, the current proposal uses an IP header to route the MKA PDU from the encrypting router (Key Server) to the decrypting router, and vice versa. To identify the packet as an MKA PDU, a UDP port is injected after the IP header with the IP header next protocol being UDP. The authors of this white paper propose that IEEE802.1X add an IANA-assigned UDP port for MKA over IP/UDP.

The IP header assignment is as follows:

1. The source IP address must be the loopback IP address of the encrypting router. In the case of MPLS, this is the loopback IP address being used for the MPLS protocol to establish the specific LSP to be encrypted.
   Note: There can be multiple LSPs originating on a single router destined to the same downstream router. In this case, each LSP may be identified with its own loopback address or, if multiple LSPs use the same loopback IP, the encryption SID and LSP ID used in the SCI field can uniquely identify the LSP.
2. The destination IP address must be the loopback IP address of the decrypting router. In the case of MPLS, like the encrypting router, there can be multiple loopback IP addresses on the decrypting router. To differentiate between multiple LSPs originating from the same upstream router and terminating on the same downstream (decrypting) router, or even the encryption SID and the LSP ID, the loopback IP address of the LSP being decrypted must be used as the destination IP address.

3. The UDP port can be the port uniquely assigned by IANA for MKA over UDP, or it can be uniquely assigned by the network administrator. If uniquely assigned by the network administrator, all the routers within the Layer 3 domain must use the same UDP port, which must be unique and unused within the Layer 3 domain.

The MKA packet is encoded following the UDP header in the following order:

1. Protocol version as defined by IEEE802.1X-2010, section 11.3.1
2. Packet type as defined by IEEE802.1X-2010, section 11.3.2
    a. This field must be set to EAPOL-MKA (0000 0101).
    b. No other value is supported.
3. Packet body, including packet Body Length, as defined by IEEE802.1X-2010, section 11.3.4

No other changes are required to the MKA packet body, key server selection, or any other aspects of MKA.

The SCI is set as described in section 5.4 Changes to the SCI.

The choice of UDP over TCP is simply because MKA has its own Hello Time, which is transmitted at regular intervals. This can be used in correlation with the MKA Life Time to determine the MKA liveness. As such, there is no need for guaranteed transmission over TCP.

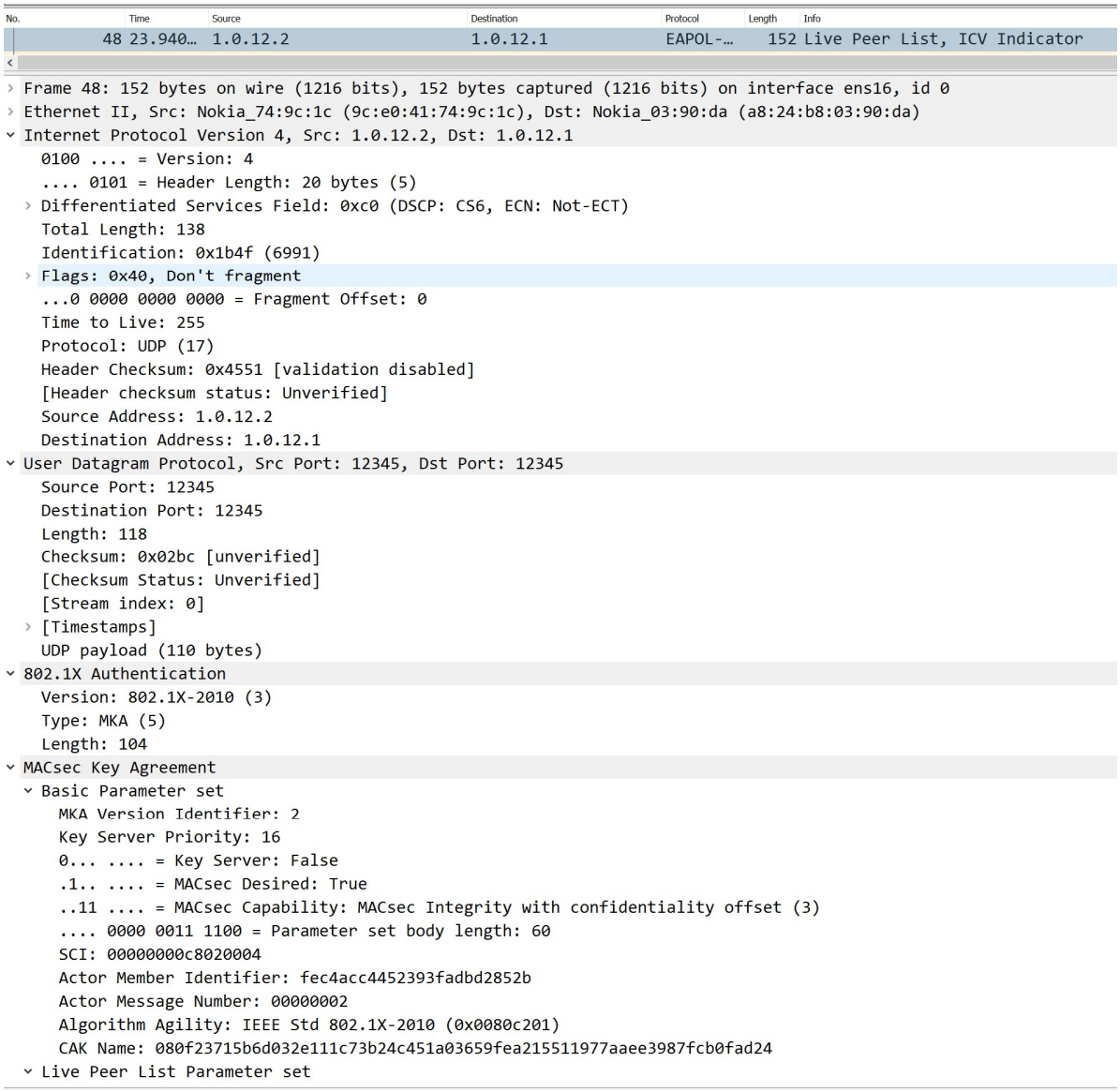
6.2 MKPDU example

The following is an example of MKA for the encrypted packet example in section 5.7 Example MPLS packet encrypted via the MACsec .

1. This example is for an LSP tunnel established from loopback IP address 1.0.12.2 to loopback IP address 1.0.12.1.
2. The LSP is a Segment Routing (RFC 8402) LSP.
3. The LSP was established via ISIS extensions for Segment Routing (RFC 8667).
4. The administrator chose a random UDP port 12345 to identify MKA over UDP/IP PDUs.
5. After UDP the EAPOL protocol version is encoded. This allows the same application code that handles MKA over EAPOL for L2 MACsec to be utilized for MKA over UDP/IP as well. This means minimal code change is required.
6. The SCI in the MKPDU is LSP-ID + Encryption SID.

7. The unique encryption SID is configured by the administrator as part of the connectivity association and advertised as part of the SCI in the MKPDU. As mentioned, this encryption SID uniquely identifies the encryption source of this LSP with in the Layer 3 network. The encryption SID is value HEX 0xC802.
8. All other fields and concepts of the MKA are in par with IEEE802.1X-2010.

Figure 12 MKA over IP/UDP used for SAK distribution for encrypting a MPLS flow



7 Conclusion

With the new developments in chip technology, both the encryption engines and the IP/MPLS forwarding plane are integrated into a single ASIC. This makes it possible to use the MACsec encryption engine to encrypt the payload of multiple OSI layer packets, for Layer 2.5 MPLS and Layer 3 IP including SRv6. The suggested updates to IEEE802.1AE standard make it possible to optimize the MACsec encryption engine to achieve encryption for different OSI layers, and enable future chip technologies to incorporate necessary changes to ensure optimal incorporation of the MACsec encryption engine with IP/MPLS protocols.

Appendix A, Deployment Examples

The following sections provides possible encryption methods for tunnel and services deployments.

As it can be seen by examples below the technology enables a flexible range of encryption over tunnels or services depending to the provider's need. The Layer 2.5 (MPLS) protocols and Layer 3 (IP) protocols work hand in hand to provide the right encryption and integrity offset for each scenario and program the encryption engine accordingly.

Example 1:

For Bulk MPLS Tunnel encryption where a single tunnel transports multiple customer services, the connectivity association (CA) can be programmed against the tunnel. In tunnel encryption case, the provider might want to keep the vc-label encrypted to not expose the customer service label (vc-label) to the transit network. In this case the implementation can explicitly indicate via a configuration on both peers that the vc-label is encrypted or alternatively the vc-label is in clear. The L3 protocols have the knowledge of the location of the service label and the tunnel labels within the label stack and can program the encryption engine correctly with the encryption and integrity offset.

To be clear this is an implementation choice. As an example, an implementation might choose to put the vc-labels in clear to achieve Inter autonomous System (Inter-AS) option B as per

RFC4364 “BGP/MPLS IP Virtual Private networks (VPNs)” where the vc-label needs to be in clear for the ASBR router to swap it from one AS to another AS or the implementation can choose to encrypt the vc-label for security reasons.

Note in the case of tunnel encryption, downloading the tunnel label stack to the encryption engine is sufficient enough for the system to match all flows that is traversing through that tunnel and encrypt them accordingly the vc-label does not and SHOULD NOT be part of the matching label stack as all the services traversing within the tunnel are encrypted via the same secY.

Figure 13 tunnel encryption with vc-label encrypted

Tunnel Encryption with tunnel containing multiple services and VC-label is encrypted

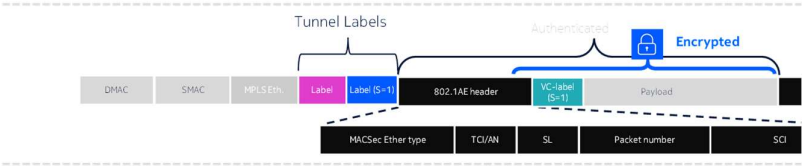
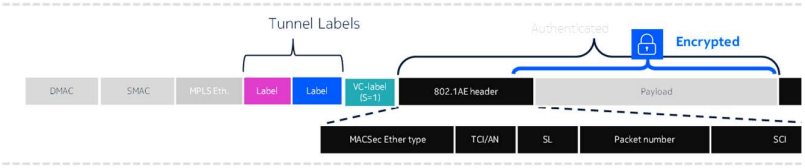


Figure 14 tunnel encryption with vc-label in clear

Tunnel Encryption with tunnel containing multiple services and VC-label is clear



Example 2:

In some cases, it might be desirable to encrypt a single service within a tunnel or encrypt each service with its own SAK which is secured with a unique PSK. This will ensure each service that is using the transport tunnel, has a different encryption key for extra security. This use case might be beneficial for providers that offer transport to multiple customers within a single tunnel, as an example smart cities or federal networks.

The system can assign a CA per service to ensure MKA is established per service. The MPLS protocols can assign the secY per label stack that uniquely identifies that service. As an example, the system can include the vc-label in the label stack which is used as the matching criteria to ensure uniqueness of label stack per service and only that service is encrypted via the assigned

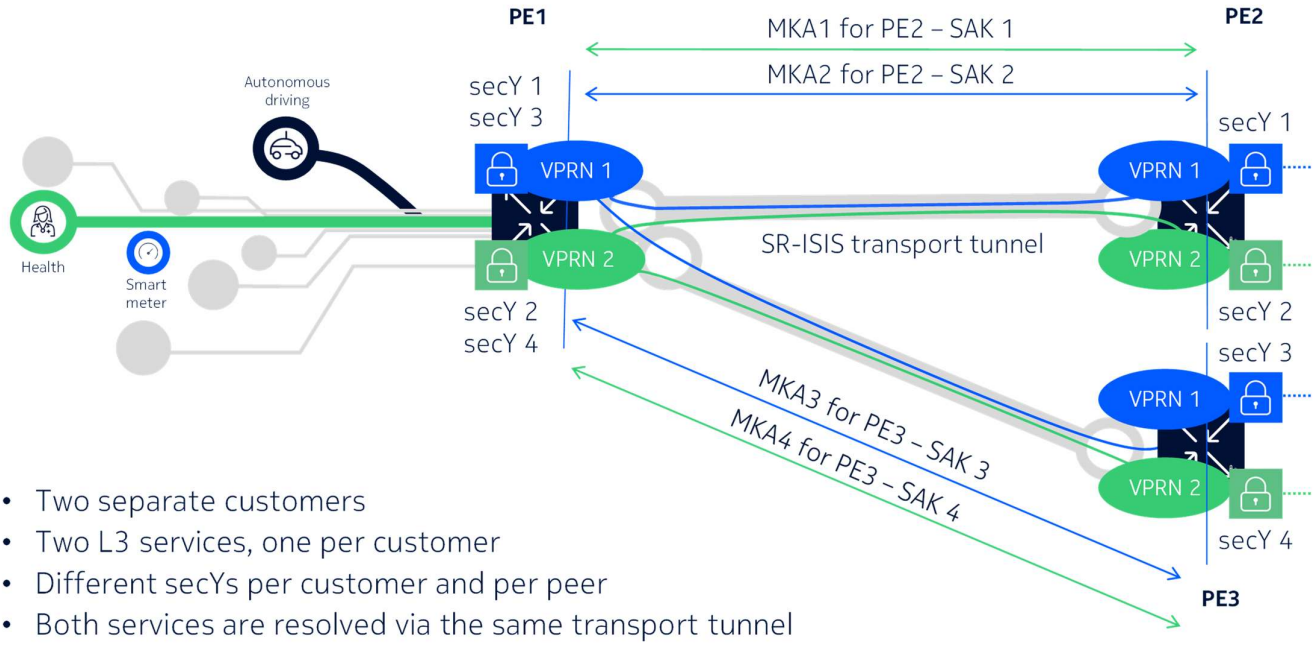
secY. After uniquely identifying the service the encryption engine can be programmed to leave the vc-label in clear or encrypted, exactly like tunnel encryption scenario.

In current implementation the MKA is established per service and per peer. This means a MKA session will be established for each service and each peer within the service distributing a unique SAK for that service and the peer.

In below figure, 2 services are configured on PE1 and its corresponding peers PE2 and PE3. As mentioned, each service will establish a MKA session for each PEER associated for that service. As an example, assuming PE1 is the key server, on PE1 (VPRN1) MKA2 is established to PE2 (VPRN1) and PE1 (VPRN1) MKA3 is established to PE3 (VPRN1), each advertising its own SAK. The same concept is followed for VPRN2.

Note: to bring up the MKA per service and per PEER is the simplest implementation currently. There might be an argument that the MKA should be a P2MP implementation, and it should be established per service only. This means all PEERs within that service will be using the same MKA and will install the same SAK for that service. This is possible as well.

Figure 15 Service encryption where each service is encrypted with its own SAK



Example 3:

Operators can combine the service encryption and tunnel encryption together for better scale. As an example, the entire tunnel can be encrypted via secY1 but a single service within that tunnel be encrypted via secY2. This type of deployment might be needed where a single customer1 has high security needs and insists a unique key is used for encrypting its traffic while other customers are good with bulk encryption. To achieve this the traffic on the tunnel is compared to matching criteria (label stack <tunnel label + service label>) of the customer1's service first to encrypt the service traffic within the tunnel with secY2 and any traffic with in the tunnel that doesn't match the service signature can be matched against the label stack of the tunnel (label stack <tunnel label>) and be encrypted via the tunnel secY1.

As an example:

1. imagine a tunnel with label1 and label2.
2. Imagine this tunnel has 3 services
    a. Service 1 is identified via vc-label label10
    b. Service 2 is identified via vc-label label11
    c. Service 3 is identified via vc-lable label12
3. The operator wants service 3 to be encrypted via its own secY2 and the rest of the tunnel (service 1 and service 2) be encrypted via a separate secY1
4. The operator can assign CA2 to service 3 and CA1 to the tunnel
5. As such the chip will be programmed with matching criteria
    a. <label1, label2, label12> →secY2
    b. <label1, label2> →secY1
6. The vc-label can be encrypted or in clear as was explained in example 1 and 2

As it can be seen this is a very robust and powerful solution as the L2.5 and L3 protocols can work hand in hand with the macsec engine to uniquely identify any flow (tunnel or service) to be encrypted via its own key and use its own MKA session for key distribution.