ANALOG DEVICES

Automotive MKA v4: Explicit SSCI

Isaac Molina
Guillermo Oliver

Problem statement

A typical automotive industry target for starting MACsec with MKA is < 50 ms.

- MKA does not achieve this by default (8s worst case).

Requires support for 10BASE-T1S shared medium where it is possible to have more than 8 peers.

MKA makes sense for Automotive MACsec with some minor modifications

- Hardcoded Key Server.
- Timing optimizations.

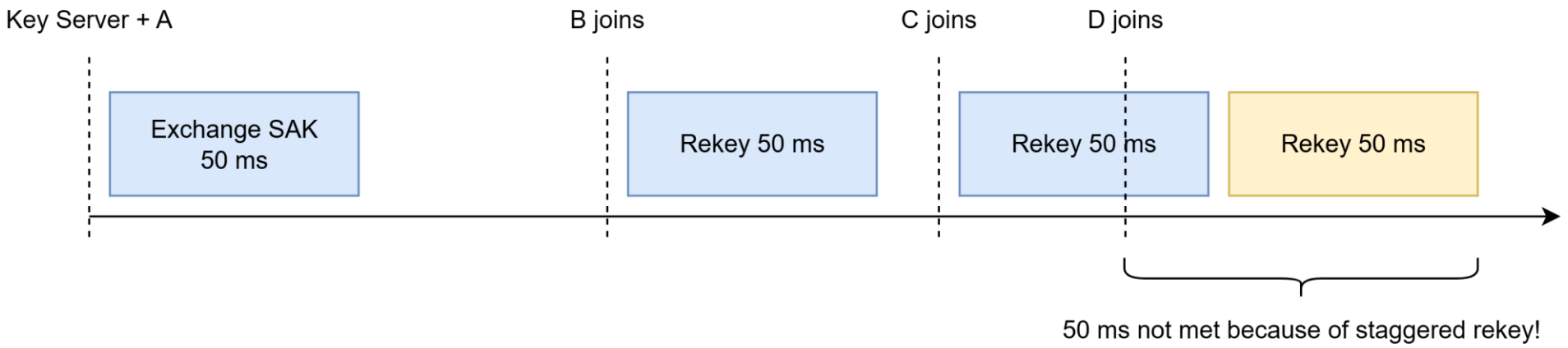
Automotive peers can go to sleep and wake-up all at the same time.

- Every time a peer joins the CA, it requires a re-key operation.

Problem statement

MKA 802.1X-2020 requires a re-key operation when a peer joins the CA and is optional when a peer leaves the CA.

- Needed to avoid the reuse of the key, Salt, and PN for security reasons.
- Constantly generating new keys adds overhead, slows thing down, and can cause delays in communication.



Can we avoid rekeying everyone when somebody joins?

Introduction

This presentation explores a novel way of managing SSCI when using XPN cipher suites to avoid the staggered rekey scenario in automotive use cases.

By explicitly assigning a unique SSCI to each participant, the Key Server can safely reuse the same key while keeping each participant's encryption space separate.

The following slides contain the specific changes proposed to various clauses of the IEEE 802.1X specification.

3. Definitions

Change the definition of Short Secure Channel Identifier of 3 as follows:

Short Secure Channel Identifier (SSCI): A 32-bit value that is unique for each SCI within the context of all SecYs using a given SAK.

NOTE—IEEE Std 802.1AEbw-2013 specified the calculation of implicit SSCI and Salt values used by the IEEE Std 802.1AE GCM-AES-XPN Cipher Suites from other MKA values. IEEE Std 802.1Xck-2018 constrained the order of entries in the MKPDU Live Peer List to facilitate that calculation.

9.8 SAK generation, distribution, and selection

Change 9.8 as follows:

If the Current Cipher Suite is not using extended packet numbering, the Key Server observes the Key Identifier and Lowest Acceptable PN for the most recent SAK in use, as transmitted by each CA member (the LKI and LLPN if LRX is true, and the OKI and OLPN otherwise; 9.10.1), and shall distribute a fresh SAK (subject to the constraints specified in 9.5 and this clause) if that Key Identifier matches the KI of the key most recently distributed and that Lowest Acceptable PN equals or exceeds the constant PendingPNExhaustion. PendingPNExhaustion is 0xC000 0000 for 32-bit PNs and 0xC000 0000 0000 0000 for 64-bit PNs. Subject to conditions [a) through c), below] that limit the frequency of SAK changes, postponing their generation and distribution until CA membership is likely to be stable, the Key Server shall also distribute a fresh SAK whenever a member is added to the live membership of CA (as perceived by the Key Server—with each MI, not the associated SCI, representing each member) unless it is sharing the explicit SSCI information, in which case, it can decide to distribute a fresh SAK or use the previously distributed SAK, and can distribute a fresh SAK when a member is removed from the live membership. If the Key Server has not generated a fresh SAK since the last addition to the Key Server's Live Peer List, an SAK is not distributed in transmitted MKPDUs. A fresh SAK is not distributed until:

9.8 SAK generation, distribution, and selection

Insert paragraph at 9.8 as follows:

If using Explicit SSCI, the Key Server observes the next explicit SSCI, and shall distribute a fresh SAK (subject to constrains specified in 9.5 and this clause) including fresh SCCI assignments, if the next explicit SSCI value equals or exceeds the constant ExplicitSSCIExhaustion. ExplicitSSCIExhaustion is 0xFFFF 0000.

9.8 SAK generation, distribution, and selection

Change 9.8 as follows:

An MKA participant does not retain any record of SAKs used prior to initialization or re-initialization, and uses a fresh MI whenever it is initialized~~, thus forcing distribution of a fresh SAK whenever it has no record of prior SAK use~~. The Key Server can choose to distribute the previously generated SAK iff using Explicit SSCI or to distribute a fresh SAK. An MKA participant is re-initialized, or deleted and a fresh participant created, if the associated SCI is changed. An MKA participant accepts only SAKs distributed by Key Servers that are mutually live, i.e., shall not accept an SAK distributed in any MKPDU that does not contain that participant's MI and acceptably recent MN in the Live Peer List.

9.8 SAK generation, distribution, and selection

Change 9.8 as follows:

Once a Key Server has distributed an SAK, it shall not distribute any previously generated SAK in any subsequent MKPDU, unless Explicit SSCI is used. However, it is recognized that successive SAKs will have the same value with a probability of no less than 1 in 2-keysize when generated as specified (see 9.8.1).

9.10 SAK installation and use

Change 9.10 as follows:

When MKA is used in conjunction with an XPN Cipher Suite, an SSCI is required for each SA. When Explicit SSCI is used, the Key Server distributes the SSCI value assigned to each SA during the SAK distribution process. Otherwise, when implicit SSCI is used, the KaY assigns SSCI values a~~. A~~s specified in IEEE Std 802.1AE, the SA with the numerically greatest SCI uses the SSCI value 0x00000001, that with the next to the greatest SCI uses the SSCI value 0x00000002, and so on. The value 0x00000000 is not used.

9.10 SAK installation and use

Insert at the end of 9.10 as follows:

On receipt of a Version 4 or higher MKPDU distributing a SAK for an XPN Cipher Suite and containing Explicit SSCI information from the Key Server, a PAE implementing version 4 determines SSCI values as follows. The Key Server SSCI is always 0x00000001. The Explicit SSCIs assignments are transmitted by the Key Server within the Explicit Live Peer List parameter set.

9.20 Explicit SSCI

Insert the following text after of 9.19:

In some environments, it is desirable to avoid distributing fresh SAK's every time a new peer joins a CA without disturbing the secure associations already established. Distributing a fresh SAK's upon a peer joining the CA guarantees that each protected MACsec message uses a unique IV value.

When using XPN cipher suites, the IV value of protected MACsec messages depends on the SSCI. If SSCI values are not reused during the life time of a SAK by new peers, the Key Server can guarantee uniqueness of IV's even when not distributing a fresh SAK.

Explicit SSCI can be used by the Key Server to distribute the active SSCI assignments to new peers, while also allocating a unique SSCI to each new peer. This is an alternative to the implicit SSCI assignation algorithm defined in clause 9.10.

A participant that receives a Explicit Live Peer List parameter set enables the Explicit SSCI functionality.

9.20 Explicit SSCI

Insert the following text after 9.20 (continue):

When the Key Server is set to use Explicit SSCI, it assigns an SSCI to each peer present in its Live Peer List and adds this information together to the Explicit Live Peer List parameter set. This parameter set replaces the Live Peer List parameter set when transmitted in a MKPDU containing a Distributed SAK parameter set.

Every time the Key Server adds a new peer to its Live Peer List, it is assigned a new SSCI. The Key Server is always assigned an SSCI of 0x00000001 and keeps track of used SSCI by using a counter that starts at 0x00000002. Each time a peer is added, the counter value SSCI is assigned to the given peer and is incremented by 1.

9.20 Explicit SSCI

Insert the following text after 9.20 (continue):

A participant that joins a CA where its Key Server is using Explicit SSCI, is mandated to initially transmit the first SAK Use parameter set indicating it can not receive data but it can transmit, until it has successfully received an MKPDU from each peer, containing both a SAK Use parameter set and an XPN parameter set. This information is required to create the peer's secure association and set the lowest acceptable packet number.

A participant upon the reception of a SAK Use parameter set from a new participant that can transmit but not receive transmits a new MKPDU as soon as possible containing its own SAK Use and XPN parameter sets.

11.11.3 Encoding MKPDUs

Change the first paragraph of 11.11.3 as follows:

c) An implementation that transmits MKPDU PDUs with an MKA Version Identifier of 1, 2, ~~or~~ 3, or 4 shall encode the protocol parameters provided by the KaY as follows:

11.11.3 Encoding MKPDUs

Change foruth paragraph of 11.11.3 as follows:

c) If there are one or more Live Peers, their Member Identifier, Message Number tuples are encoded within a Live Peer List as specified in Figure 11-9. When an implementation transmits MKPDUs with an MKA Version Identifier of 4, a Key Server using Explicit SSCI replaces the Live Peer List parameter set with the Explicit Live Peer List parameter set specified in Figure 11-X, which includes the SSCI for each peer, when transmitting an MKPDU that contains a Distributed SAK parameter set. An implementation that transmits MKPDUs with an MKA Version Identifier of 3 or 4 shall order the entries in the Live Peer List and shall encode the Key Server's SSCI in Octet 2 of the Live Peer List parameter set of MKPDUs containing a Distributed SAK parameter set for use with an XPN Cipher Suite, as specified in 9.10.

11.11.4 Decoding MKPDUs

Change 11.11.4 as follows:

b) Each of the following parameter sets, if any, shall be identified by its parameter set type and decoded as specified in Table 11-7, provided that the set is completely present (as indicated by its body length parameter) within the MKPDU prior to the ICV and the parameter set body length is:

1) A multiple of 16 octets, if the parameter set is a Live or Potential Peer List.

2) 0, 40, or more octets, if the parameter set is the MACsec SAK Use parameter set.

3) 0, 28, 36, or more octets, if the parameter set is the Distributed SAK parameter set.

4) 28 or more octets, if the parameter set is the Distributed CAK parameter set.

5) 5 or more octets, if the parameter set is the KMD parameter set.

6) A multiple of 20 octets, if the parameter set is the Explicit Live Peer List parameter set.

Explicit Live Peer List parameter set

Insert the following paragraph before of 11.3:

Bit:	8	7	6	5	4	3	2	1	Octet:
Parameter set type = 9									1
X	X	X	X	X	X	X	X		2
X	X	X	X	Parameter set body length					3
Parameter set body length (cont)									4
Member Identifier									5-16
Message Number									17-20
SSCI									21-24
Member Identifier									a
Message Number									
SSCI									

[a]Member Identifier, Message Number, SSCI tuples are repeated to the end of the parameter set.

Comparison with KN rollover

The Key Number rollover proposal described at: https://www.ieee802.org/1/files/public/docs2025/x-seaman-mka-optimizations-0325-v06.pdf section 6.4.

- Solves the problem of performing the necessary calculations to unwrap and install the SAK.
- **Does not prevent the need to create a new SA with the new KN even if the SAK is the same as previously distributed and still requires the Key Server to coordinate the transmit rollover for all participants to transition from CP:READY to CP:TRANSMIT in the CP state machine (three way handshake).**
- Peers implementing this proposal can interoperate with other peers that do not implement it, although it changes the rules for fresh SAK use specified in 9.8 of 802.1X.
- In this proposal, it is highlighted the importance of not prolonging the life time of a SAK, as it makes easier cryptanalytic attacks.

Explicit SSCI

The Explicit SSCI proposal described in this slide deck:

- Solves the problem of performing the necessary calculations to unwrap and install the SAK.
- Avoids the need to create a new SA, and performing the SAK rollover (three way handshake) when a new participant is added.
- Key Numbers are only incremented upon a fresh SAK distribution, creating a distinct fraction of the nonce space for each of the 2^{32} Key Numbers used with its MI.

- The negative points are that:
  - Can only be used if all Live Peers are using MKA version 4.
  - The absence of any convenient mechanism for informing the Key Server that the peer does not want to use Explicit SSCI.

Proposal comparison

	MKA v3	KN rollover	Explicit SSCI
SAK unwrap	Yes	No	No
SAK rollover	Yes	Yes	No
Changes to 802.1AE	N/A	No	Minor (Editorial)
Changes to 802.1X	N/A	Minor	Requires a new parameter set for Explicit Live Peer List
Replay protection	Yes	Yes	Yes

Conclusion

The Explicit SSCI proposal addresses the key rollover bottleneck in automotive and other applications that need a very fast startup, when multiple peers try to join a CA closely together, avoiding successive key rollovers, one after another.

By introducing Explicit SSCI handling for XPN cipher suites, we can distribute the same SAK's to new peers without compromising its security.

This approach reduces unnecessary key updates, and allows faster group formation times with minimum modifications to the MKA specification.

Bibliography

MKA optimization for group CAs, Mick Seaman, March 2025

https://www.ieee802.org/1/files/public/docs2025/x-seaman-mka-optimizations-0325-v06.pdf

https://www.ieee802.org/1/files/public/docs2025/x-seaman-mka-optimize-ppt-0305-v01.pdf

Automotive Requirements for MKA, Dr. Lars Völker, March 2025

https://www.ieee802.org/1/files/public/docs2025/x-voelker-mka-requirements-automotive-0325-v01.pdf

AHEAD OF WHAT'S POSSIBLE

analog.com

ANALOG DEVICES