STARTING UP MACSEC FOR AUTOMOTIVE ETHERNET.

7th International VDI Conference – Cyber Security for Vehicles.

Dr. Lars Völker

2021-06-22

TECHNICA ENGINEERING

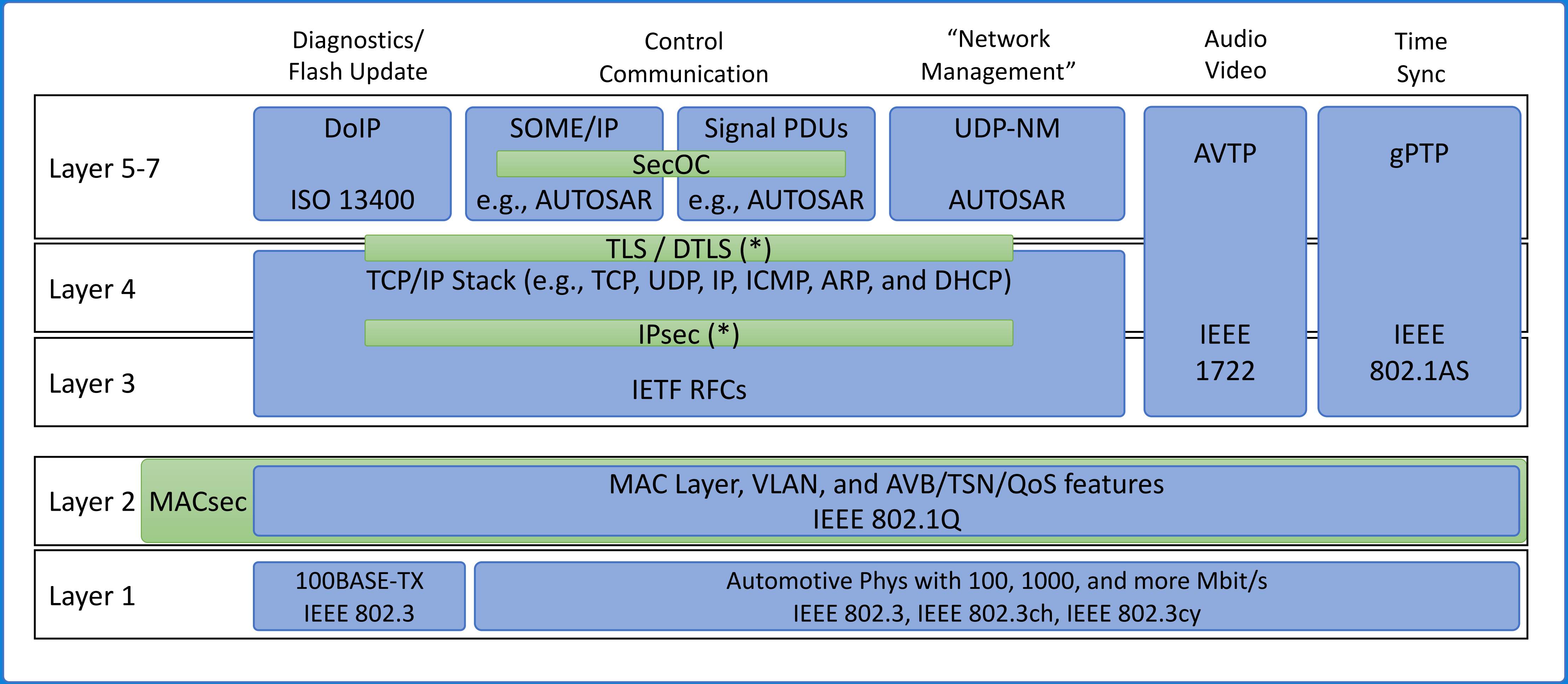
STARTING UP MACSEC

TABLE OF CONTENTS

technica engineering

1 CHAPTER.
MACSEC INTRO.

SIMPLIFIED COMMUNICATION STACK.



(*) Typically unicast only.

WHY IS MACSEC SO INTERESTING?

1
- MACsec is currently the only solution that can protect all communication on Automotive Ethernet against external attackers.
  - Alternatives (e.g., IPsec, (D)TLS, SecOC) leave many protocols unprotected.

2
- MACsec can protect Multicast and Broadcast communication.
  - Better than (D)TLS and regular IPsec.

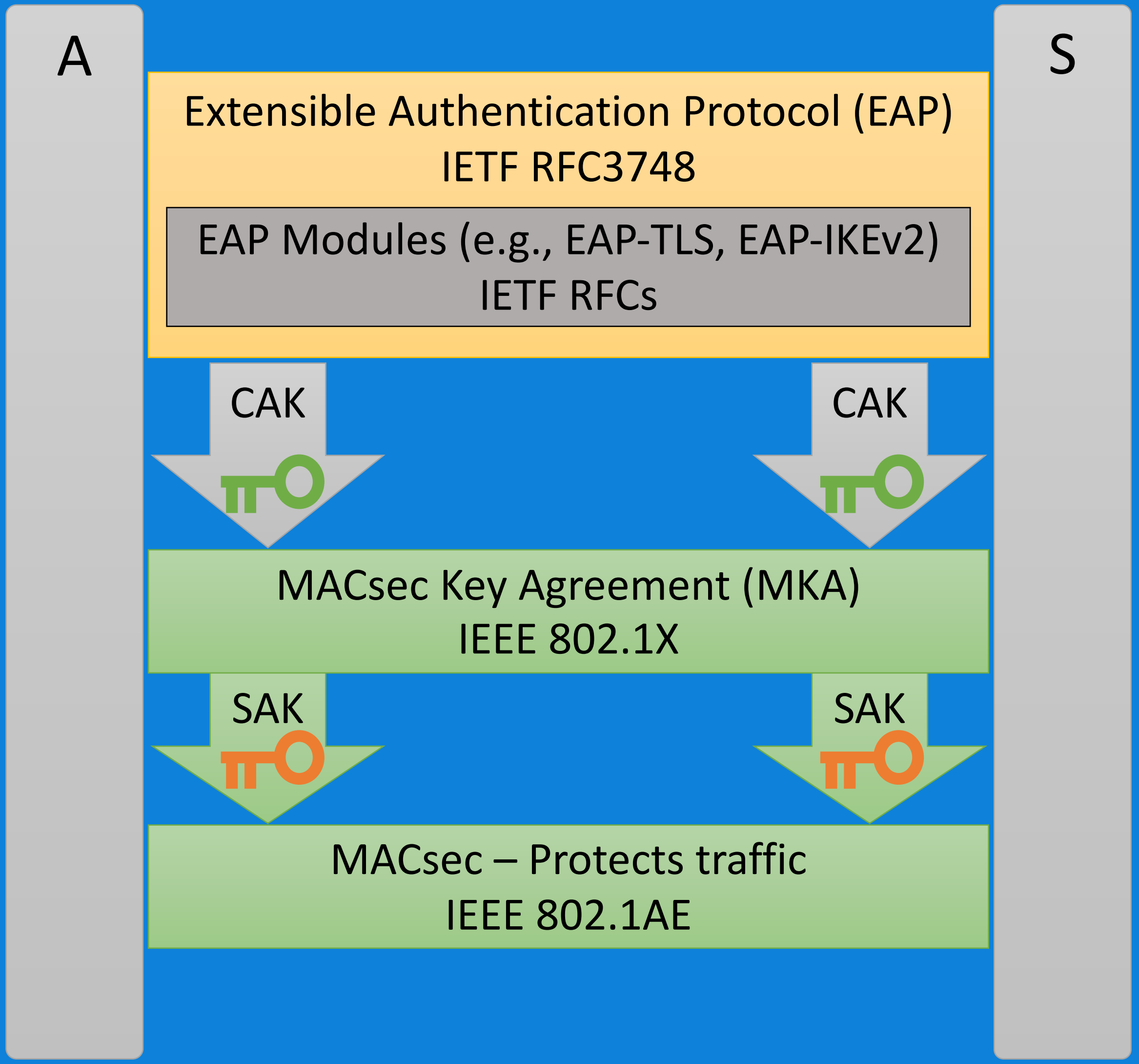
3
- MACsec can protect all traffic on a link with one association.
  - Less keys and key exchanges required (better than SecOC, (D)TLS, IPsec).

4
- MACsec can be run hop-by-hop:
  - You don't need to share keys for large groups (better than SecOC).

For further details see:
Dr. Lars Völker, BMW: "Comparing Automotive Network Security for Different Communication Technologies", Automotive Ethernet Congress, 2018.

MACSEC KEY HIERARCHY.

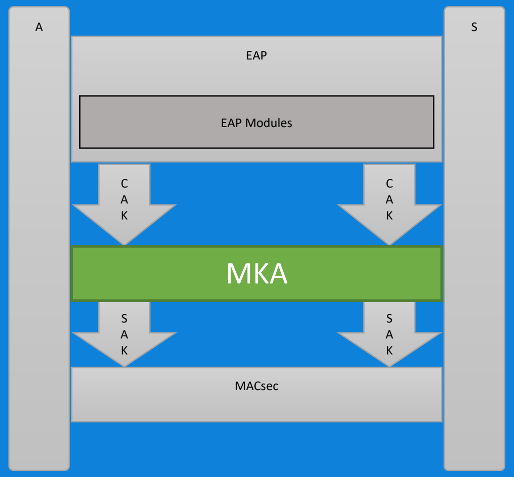
- EAP:
  - The Authenticator (A) controls access of the Supplicant (S).
- EAP modules:
  - Authenticate and authorize supplicant.
  - Agree on Connectivity Association Key (CAK).
  - E.g., EAP-TLS, EAP-IKEv2.
- MACsec Key Agreement (MKA):
  - Distribute Secure Association Key (SAK).
  - Monitoring packet numbers.
  - Rekeying.
- MACsec:
  - Protect communication (auth. or auth.+enc.).



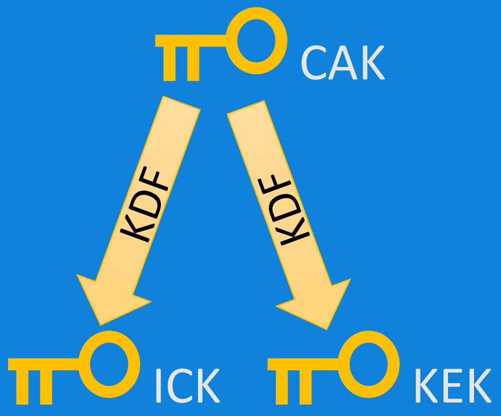
technica engineering

MKA OVERVIEW.

- Communication partners have the same secret CAK.
- Additional keys are derived via an AES-CMAC KDF:
  - ICV Key (ICK): MKA message integrity protection (AES-CMAC).
  - Key Encryption Key (KEK): encryption of keys in MKA messages.



- Key Exchange process:
  - Find suitable peers and check their liveliness.
  - Elect key server (with EAP obvious).
  - Key server distributes SAK (encrypted by KEK using AES Key Wrap).
  - Activate SAK in MACsec.



2 CHAPTER.

KEY EX OPTIONS.

AUTHENTICATION OPTIONS.

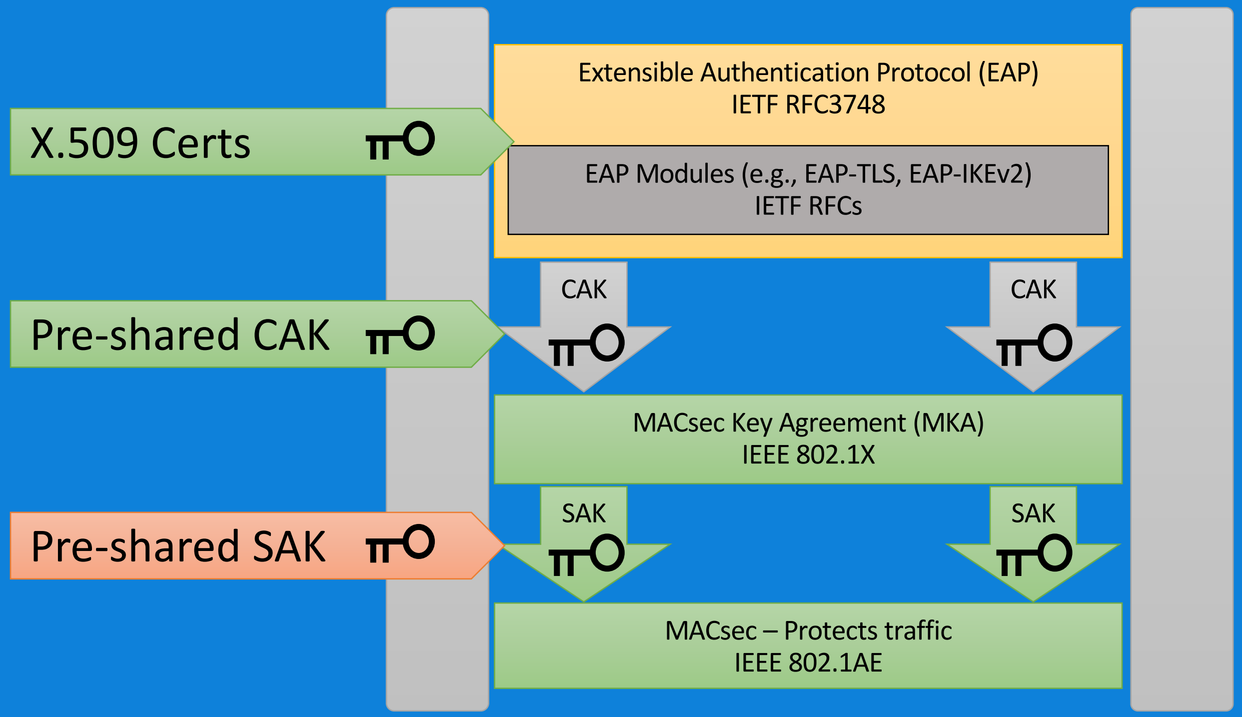
- MACsec/MKA + EAP can support almost every authentication option:
  - Passwords, PSKs, certificates, hardware tokens, …
- Current automotive options in series production:
  - Symmetric keys (e.g., for AES or hash functions).
  - Certificates (e.g., X.509).
- Aspects to keep in mind:
  - Replay attacks.
  - Fast startup requirements for automotive use cases.

KEY EXCHANGE OPTIONS.

- X.509 certificates:
  - EAP-TLS1.2 (RFC 5216).
  - EAP-TLS1.3 (currently draft only).
  - EAP-IKEv2 (RFC 5106).

- Symmetric keys (128/256 bit):
  - Pre-shared CAKs (MKA).
  - Pre-shared SAKs (MACsec).

  → Key reuse possible! Unsecure!



technica engineering

3 CHAPTER. PERFORMANCE.

MKA OPEN-SOURCE CODE.

- Our team started with an Open-Source implementation.

- First run MKA without EAP: ~3s (sic!):

No.	Time	Time Delta	Source	Destination	Protocol	Length	Info
1	0.000000000	0.000000000	aa:ea:c4:e5:42:cc	01:80:c2:00:00:03	EAPOL-MKA	98	Key Server
2	0.986986779	0.986986779	ce:e9:55:df:c2:5e	01:80:c2:00:00:03	EAPOL-MKA	98	Key Server
3	2.001422945	1.014436166	aa:ea:c4:e5:42:cc	01:80:c2:00:00:03	EAPOL-MKA	118	Key Server, Potential Peer List
4	2.988365546	0.986942601	ce:e9:55:df:c2:5e	01:80:c2:00:00:03	EAPOL-MKA	150	Key Server, Live Peer List, Distributed SAK
5	2.995237588	0.006872042	ce:e9:55:df:c2:5e	01:80:c2:00:00:03	EAPOL-MKA	194	Key Server, Live Peer List, MACsec SAK Use, Distributed SAK
6	2.995736763	0.000499175	aa:ea:c4:e5:42:cc	01:80:c2:00:00:03	EAPOL-MKA	162	Live Peer List, MACsec SAK Use
7	2.996580117	0.000843354	aa:ea:c4:e5:42:cc	01:80:c2:00:00:03	EAPOL-MKA	162	Live Peer List, MACsec SAK Use

- Why is this so slow?
  - Both peers send with MKA Hello Time = 2s (see standard) regularly.
  - For election process, peer needs to be found and added to Live Peer List.
  - Only the MACsec SAK Use is send faster (on change).

- Assumptions of IEEE 802.1X are not fully automotive compatible:
  - IEEE 802.1X aims for a bounded time but not a performance target.

OPTIMIZATIONS.

1. Optimize send timings.

- For the peers to find each other, peers should send more frequently.
- Slow down when SAK is established or in Live Peer List of Key Server.

2. Configure Key Server priority.

- With PSK, MKA does not assume who is key server (with EAP this is clear).
- Make sure this is configured and no peer waits for election.

3. Configure number of peers (“1” in hop-by-hop mode).

- MKA does not assume number of peers; thus, it waits.
- Key Server can generate key as soon as “1” peer is in its Live Peer List.

4. ICK and KEK can be precalculated and securely stored to save time.

- Many stacks calculate the AES Key Wraps at startup, but HSM might be busy.

EXAMPLE: EAP-TLS + OPTIMIZED MKA.

No.	Time	Time Delta	Source	Destination	Protocol	Length	Info
1	0.000000000	0.000000000	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	23	Request, Identity
2	0.002054584	0.002054584	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	31	Response, Identity
3	0.003316137	0.001261553	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	24	Request, TLS EAP (EAP-TLS)
4	0.007923225	0.004607088	52:54:00:aa:62:b6	01:80:c2:00:00:03	TLSv1.2	208	Client Hello
5	0.011259959	0.003336734	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	1421	Request, TLS EAP (EAP-TLS)
6	0.012566842	0.001306883	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	24	Response, TLS EAP (EAP-TLS)
7	0.013733349	0.001166507	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	1421	Request, TLS EAP (EAP-TLS)
8	0.014088112	0.000354763	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	24	Response, TLS EAP (EAP-TLS)
9	0.014999081	0.000910969	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	TLSv1.2	786	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
10	0.020397395	0.005398314	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	1426	Response, TLS EAP (EAP-TLS)
11	0.021962444	0.001565049	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	24	Request, TLS EAP (EAP-TLS)
12	0.022412434	0.000449990	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	1422	Response, TLS EAP (EAP-TLS)
13	0.023837778	0.001425344	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	24	Request, TLS EAP (EAP-TLS)
14	0.024133873	0.000296095	52:54:00:aa:62:b6	01:80:c2:00:00:03	TLSv1.2	513	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
15	0.026230843	0.002096970	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	TLSv1.2	75	Change Cipher Spec, Encrypted Handshake Message
16	0.026966196	0.000735353	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	24	Response, TLS EAP (EAP-TLS)
17	0.027921076	0.000954880	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	22	Success
18	0.045348454	0.017427378	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	82	
19	0.047968715	0.002620261	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	82	Key Server
20	0.048169985	0.000201270	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	102	Key Server, Potential Peer List
21	0.048263792	0.000093807	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	102	Potential Peer List
22	0.048546117	0.000282325	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	102	Live Peer List
23	0.049108479	0.000562362	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	134	Key Server, Live Peer List, Distributed SAK
24	0.049753040	0.000644561	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	178	Key Server, Live Peer List, MACsec SAK Use, Distributed SAK
25	0.049777575	0.000024535	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	146	Live Peer List, MACsec SAK Use
26	0.050140095	0.000362520	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	146	Live Peer List, MACsec SAK Use

28ms

22ms

EAP-TLS

Fragmented!

Fragmented!

MKA

- Key Exchange: ~50ms (with first but not all proposed code optimizations).
  - EAP + EAP-TLS: 28ms (including certificate chain transports).
  - MKA: < 22ms (including 17ms wait times).
  - EAP-TLS, TLS 1.2, ECDH, Certificate chains transported (3k).

EXAMPLE: EAP-IKEV2 + OPTIMIZED MKA.

No.	Time	Time Delta	Source	Destination	Protocol	Length	Info
1	0.000000000	0.000000000	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	23	Request, Identity
2	0.000774654	0.000774654	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAP	33	Response, Identity
3	0.007623369	0.006848715	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	ISAKMP	272	IKE_SA_INIT MID=00 Initiator Request
4	0.012049713	0.004426344	52:54:00:aa:62:b6	01:80:c2:00:00:03	ISAKMP	336	IKE_SA_INIT MID=00 Responder Response
5	0.019149714	0.007100001	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	ISAKMP	144	IKE_AUTH MID=01 Initiator Request
6	0.021785272	0.002635558	52:54:00:aa:62:b6	01:80:c2:00:00:03	ISAKMP	144	IKE_AUTH MID=01 Responder Response
7	0.026723725	0.004938453	52:54:00:5c:f9:b1	52:54:00:aa:62:b6	EAP	22	Success
8	0.030178398	0.003454673	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	82	Key Server
9	0.036720458	0.006542060	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	82	
10	0.037085717	0.000365259	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	102	Key Server, Potential Peer List
11	0.039702837	0.002617120	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	102	Potential Peer List
12	0.040614892	0.000912055	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	134	Key Server, Live Peer List, Distributed SAK
13	0.041910897	0.001296005	52:54:00:5c:f9:b1	01:80:c2:00:00:03	EAPOL-MKA	178	Key Server, Live Peer List, MACsec SAK Use, Distributed SAK
14	0.047462890	0.005551993	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	146	Live Peer List, MACsec SAK Use
15	0.053748876	0.006285986	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	146	Live Peer List, MACsec SAK Use
16	0.055796143	0.002047267	52:54:00:aa:62:b6	01:80:c2:00:00:03	EAPOL-MKA	146	Live Peer List, MACsec SAK Use

27ms

EAP-IKEV2

29ms

MKA

- Key Exchange: ~56ms (with first but not all proposed code optimizations).
  - EAP + EAP-IKEv2 (no certs): 27ms (but no certificates transported).
  - MKA: 29ms (including 3.5ms wait time before MKA starts).
  - EAP-IKEv2, DH, no certificate chain transported (not realistic).
- Even after tuning MKA code, results still not stable!

Slow answers of peer (not Key Server).

technica engineering

FIRST RESULTS.

- After first optimizations: MKA runs in <30ms.
  - MKA timings fluctuate a lot: best cases are <5ms (without wait time).
- ~30ms for certificate-based authentication (EAP-TLS and EAP-IKEv2).
- Platforms (experiments on Raspberry Pi):
  - On a real ECU the asymmetric operations will take longer!
  - Certs: 1 ECDH + 1 ECDSA-sign + (n) ECDSA-verify (n certs in chain).
  - MKA itself should be very fast on embedded ECUs due to AES acceleration.
- Additional optimizations possible.

4 CHAPTER. SUMMARY.

SUMMARY.

- Symmetric keys: static CAK with MKA only → recommended!
- Certificates: EAP-IKEv2 or EAP-TLS (1.3 if possible).
  - Tune algorithm selection.

- MKA uses only AES operations, which can use accelerators.
- Tune the MKA implementations based on automotive assumptions!

- Other recommendations:
  - Choose MACsec algo (e.g., GCM-AES-256 with XPN) and rekey settings.
  - Add mechanisms (e.g., filters) to counter internal attackers too.

STAY CONNECTED

Dr. Lars Völker

Technical Fellow
Lars.Voelker@technica-engineering.de
+49 (0) 175 1140982

Technica Engineering GmbH
Leopoldstraße 236
80807 Munich
Germany

This presentation would not have possible without contributions of our awesome MACsec team!

Andreu, Johan, Jordi, Carlos, Marc, and Ramon of our hardware and software development center in Barcelona for building great prototypes and laying the groundwork for A samples supporting MACsec.

Jose, Manuel and Antonio of our Munich Security group who lead the team in the right direction and create OEMs specifications.

Our semiconductor vendors supplying insights and early silicon in these challenging times.

And finally, customers for trusting us.

Thank you all!