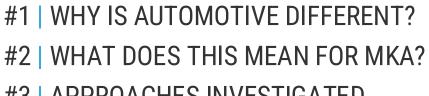


AUTOMOTIVE REQUIREMENTS FOR MKA

Dr. Lars Völker

AUTOMOTIVE REQUIREMENTS FOR MKA TABLE OF CONTENT



#3 | APPROACHES INVESTIGATED.





#1 AUTOMOTIVE REQUIREMENTS IN GENERAL

WHY IS AUTOMOTIVE DIFFERENT? INTRODUCTION



- Modern vehicles are becoming moving Ethernet networks.
 - Automotive PHYs: 100BASE-T1, 1000BASE-T1, 10G/5G/2.5GBASE-T1, 25GBASE-T1.
 - Also: 10BASE-T1S supporting shared medium for low-cost use cases
- Automotive OEMs produce many Ethernet networks per minute.
 - Fully automated production processes (e.g., key and certificate install) needed.
- Lots of multicast communication.
 - That's why many OEMs are introducing MACsec.
- Lots of safety critical and real-time communication.
- Vehicle startup performance is very critical!

WHY IS AUTOMOTIVE DIFFERENT? VEHICLE STARTUP IS CRITICAL.



- How long should your car take to be ready to drive?
 - Typical target to achieve type approval: 1.5s 2s (from touching car to driving off).
 - Critical features must work already (driving, breaking, steering, ...)
 - Mandatory Backup Cameras (e.g., US) needs to work, which may use Ethernet.
- 1.5 2s sounds like a lot but you need to:
 - Detect car is being unlocked
 - Power-Up/Wake-up and boot Electronic Control Units (ECUs) think small computers.
 - Start stacks, middleware, applications, etc.
 - Get Ethernet links up (that's why T1 PHYs link up in 20-120ms).
 - Switches need to start too.
 - Start Secure Communication (e.g., MACsec for all links).

WHY IS AUTOMOTIVE DIFFERENT? EVEN WORSE: STARTUP HAPPENS OFTEN...



- For power saving (= fuel saving) reasons, ECUs may go to sleep
 - And wakeup again, when needed --> think another startup
- Undervoltage when cranking a combustion engine
 - Multiple ECUs are knocked out and restart after a very short time.
 - Very fast restart needed as you need to drive soon.
 - Restarting your car on train tracks as a train gets closer...
- Startup happens very often and is highly critical!

WHY IS AUTOMOTIVE DIFFERENT? LIFETIME AND COST



- Automotive needs to design for a long lifetime
 - Parts need to be available e.g., 10 years after end of production.
 - Electronics need to last longer than in IT.
 - E.g., flash wears over time, so we cannot just write to flash all the time.
- Environmental
 - ECUs need to work in for example -40°C to +125°C (-40 F to 257 F)
 - Higher temperatures zone exist.
- Cost is also very critical as so many ECUs are needed.

The x - False od.use_y = True od.use_z = False mation == "MIRROR_Z": mod.use_x = False mod.use_y = False mod.use_z = True

groups_free(struct group_info *group_info)
informing if (groupinfo->blocks[0] != group_info->small_block) { for (i = 0; i < group_info->nblocks; i++)
int i; freepage((unsigned long)groupinfo->blocks[i]);
for (i = 0; i < group_info->nblocks; i++) freepage((unsigned long)groupinfo->blocks[i]); kfree(groupinfo) kfree(groupinfo);

(groupsalloc);

roups_free(struct group_info *group_info)

HBOL(groupsalloc);

active = modifier of property (groupsfree); select= 1 EXPORTSYMBOL(groupsfree);

ob.select=1 /* export the groupinfo to a user-space array */ lected" + str(modifiertach)) http:modificuser(gid t user *grouplist,

const struct group_info *group info)
groups_touser(gid_t _user *grouplist,

const struct group info *group info)

int i, unsigned int count = groupinfo->ngroups; int í;

unsigned int count = groupinfo->ngroups; for (i = 0; i < group_info->nblocks; i++) { unsigned int cpcount = min(NGROUPSPERBLOCK, count);
for (i = 0; i < group_info->nblocks; i++) { unsigned int len = cpcount * sizeof(*grouplist); unsigned int cpcount = min(NGROUPSPERBLOCK, count); unsigned int len = cpcount * sizeof(*grouplist); unsigned inc inc approach (*grouplist);
if (copyto_user(grouplist, group_info->blocks[i], len)) n -EFAULT; 1_user(grouplist, group_info->blocks[i], len))

AUTOMOTIVE #2 REQUIREMENTS **MKA SPECIFIC**

WHAT DOES THIS MEAN FOR MKA? STARTUP PERFORMANCE OF "AUTOMOTIVE MKA"



- A typical Automotive Industry target for starting MACsec with MKA: <30ms.
 - MKA does not achieve this by default (8s or less guaranteed).
- As MKA makes sense for Automotive MACsec too:
 - Hard-coding Key Server priority and optimizing timings.
 - This allows for less than 10ms startup while being compatible to MKA (*).
 - However, this only works for Switch Ethernet (2 stations).
- Along came 10BASE-T1S Shared Medium
 - And startup timing for e.g., 7 stations is a problem again.

WHAT DOES THIS MEAN FOR MKA? RECAP: MKA EXAMPLE WITH ONLY 2 STATIONS

Key Server



Distributed SAK parameter set Parameter set type: Distributed SAK (4)

01.. = Distributed AN: 1

..00 = Confidentiality Offset: No confidentiality (0)

.... 0000 0010 0100 = Parameter set body length: 36

Key Number: 00000001

MACsec Cipher Suite: GCM-AES-XPN-128 (0x0080c20001000003) AES Key Wrap of SAK: f0636a29710eaf847cea17a960e60c98cbe78b043ca44b61

MACsec SAK Use parameter set Parameter set type: MACsec SAK Use (3) 00.. = Latest Key AN: 0 ..0. = Latest Key tx: False ...0 = Latest Key rx: False 01.. = Old Key AN: 11. = Old Key tx: True1 = Old Key rx: True 0.... = Plain tx: False .0.. = Plain rx: False ...0 = Delay protect: False 0000 0010 1000 = Parameter set body length: 40 Latest Key: Key Server Member Identifier: 0000000000000000000000 Latest Key: Key Number: 00000000 Latest Key: Lowest Acceptable PN: 00000001 Old Key: Key Server Member Identifier: cccb1f0eee52ba6468a7df3e Old Key: Key Number: 00000001 Old Key: Lowest Acceptable PN: 00000001

Allows also to rekey without break communication.

Peer

Key Server#1 (KeyServer Prio.: 0)

Potential Peer List: {}

Peer#1 (KeyServer Prio.: 255)

Potential Peer List: {Key Server}

KeyServer#2

Live Peer List: {Peer}, Dist SAK, SAK Use

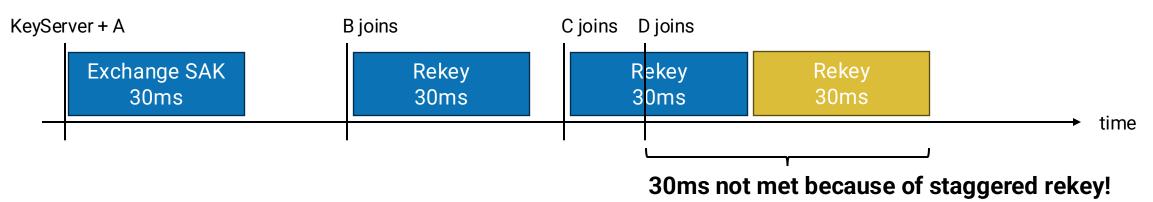
Peer#2

Live Peer List: {Key Server}, **SAK Use**

WHAT DOES THIS MEAN FOR MKA?

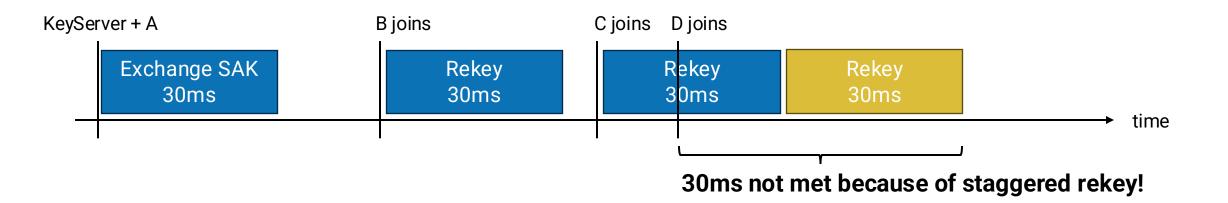


- RECAP: MKA RULES FOR STATIONS JOINING
- MKA needs to avoid reuse of key + salt + PN/XPN for security reasons.
- Also, reused PN/XPN might be detected as replay and not accept.
- MKA (see 802.1X-2020, 9.8):
 - Distribute new SAK on member joining.
 - Optional on member being removed.
- What can go wrong, if we can rekey in 30ms?



WHAT DOES THIS MEAN FOR MKA? RECAP: MKA RULES FOR STATIONS JOINING (2)





- You could break the rekey in progress, but this might drop connectivity or make C's join late!
- Plus: ECUs are faster on startup but when MACsec is running, they have higher application load and MKA performance is reduced. Minimizing their work would help.
- Can we avoid Rekeying everyone, when someone joins?



#3 AUTOMOTIVE VS. MKA APPROACHES

APPROACHES INVESTIGATED.



APPROACH 1: "DO NOT REKEY ON JOIN".

- On join: Key Server just redistributes SAK with same KN.
- On leave: Rekey.
- Does not work for XPN due to SSCI change. With 10Mbit/s acceptable!?
- How to ensure leave is detected before 6s timeout???
- With undervoltage situation, multiple stations fall out and come back.
 - It is right now not reliable to know, if a station used a SAK before going out.
 - You could have just lost a SAK Use.
 - We need to rekey on every station rejoining as we do not know what they used last.
 - This can lead to a staggered restart, which is the same as Rekey on Join.
- Overall: Degrades into the "Rekey on Join" worst case and does not help.

APPROACHES INVESTIGATED.

APPROACH 2: "KEY SERVER HOLDS STATE AND FAST FORWARDS".

- Key Server collects and holds state of all stations (SAK Use).
- If a station comes back, Key Server sends state (like SAK Use content) out with key.
 - Key Server adds a margin to PN/XPN.
- This pushes stations XPN/PN forward
- Returning station needs to check all SAK Use of other ECUs to set up RX PN/XPNs.
- Con: Incompatible MKA change.

▼ MACsec Key Agreement
▶ Basic Parameter set
Live Peer List Parameter set
MACsec SAK Use parameter set
Distributed SAK parameter set
 Experimental Fast Forward set
Parameter set type: Fast Forward (Experimental) (128) Reserved: 00
0001 = Variant: 1
0000 0101 0000 = Parameter set body length: 80
Latest Key: Key Number: 00000001
Old Key: Key Number: 00000000
▼ Peer Entry 0
Actor Member Identifier: afd7c497baa3c653de2e7824
Actor Message Number: 0000023f
SCI: 94103eb83af90001
Latest Key: Lowest Acceptable PN (32 MSB): 00000001
Old Key: Lowest Acceptable PN (32 MSB): 0000000
00 = Latest Key AN: 0 1 = Latest Key tx: True
1 = Latest Key tx: True 1 = Latest Key rx: True
00 = Old Key AN: 0
0 = Old Key rx: False
0 = Plain tx: False
.0 = Plain rx: False
0 = Delay protect: False
Reserved: 0000
Peer Entry 1
Peer Member Identifier: 596286ce57cfc4e5fd770ec6
Peer Message Number: 00000003 SCI: 0014223344550001
Latest Key: Lowest Acceptable PN (32 MSB): 00000000
Old Key: Lowest Acceptable PN (32 MSB): 00000000
00 = Latest Key AN: 0
= Latest Key tx: False
0 = Latest Key rx: False
00 = Old Key AN: 0
0. = Old Key tx: False
0 = Old Key rx: False
0 = Plain tx: False
.0 = Plain rx: False
0 = Delay protect: False
Reserved: 0000 Integrity Check Value: 9e3845631d067d3f1c70358aa1998bf7
THEER TEY CHECK VALUE: 90304303100070311070330881990017

Example for illustration only



APPROACHES INVESTIGATED.



APPROACH 3: "SSCI BUMPING"

- Idea: Make SSCI changes on join/leave compatible.
 - Key Server adds new station to the peer list without changing other SSCIs.
 - Key Server does not free SSCIs, when station is removed, but marks them invalid.
 - When station rejoins, Key Server gives out new SSCI.
- Con: XPN only solution means higher cost. Better than no solution.
- Con: Not quite compatible to MKA:
 - SSCI list not ordered anymore. Is this a problem?
 - To be defined in MKA: How to mark SSCIs as invalid, so that they are ignored (KAY).

APPROACHES INVESTIGATED. SUMMARY.



- Many approaches investigated (only showed 3 examples).
- No working solution that is fully MKA compatible was identified.
- Open questions:
 - How can we solve this?
 - Do we need to update MKA?









Technica Engineering GmbH

Leopoldstraße 236 80807 Munich Germany

DR. LARS VÖLKER

Technical Fellow lars.voelker@technica-engineering.de +49 175 11 40 982