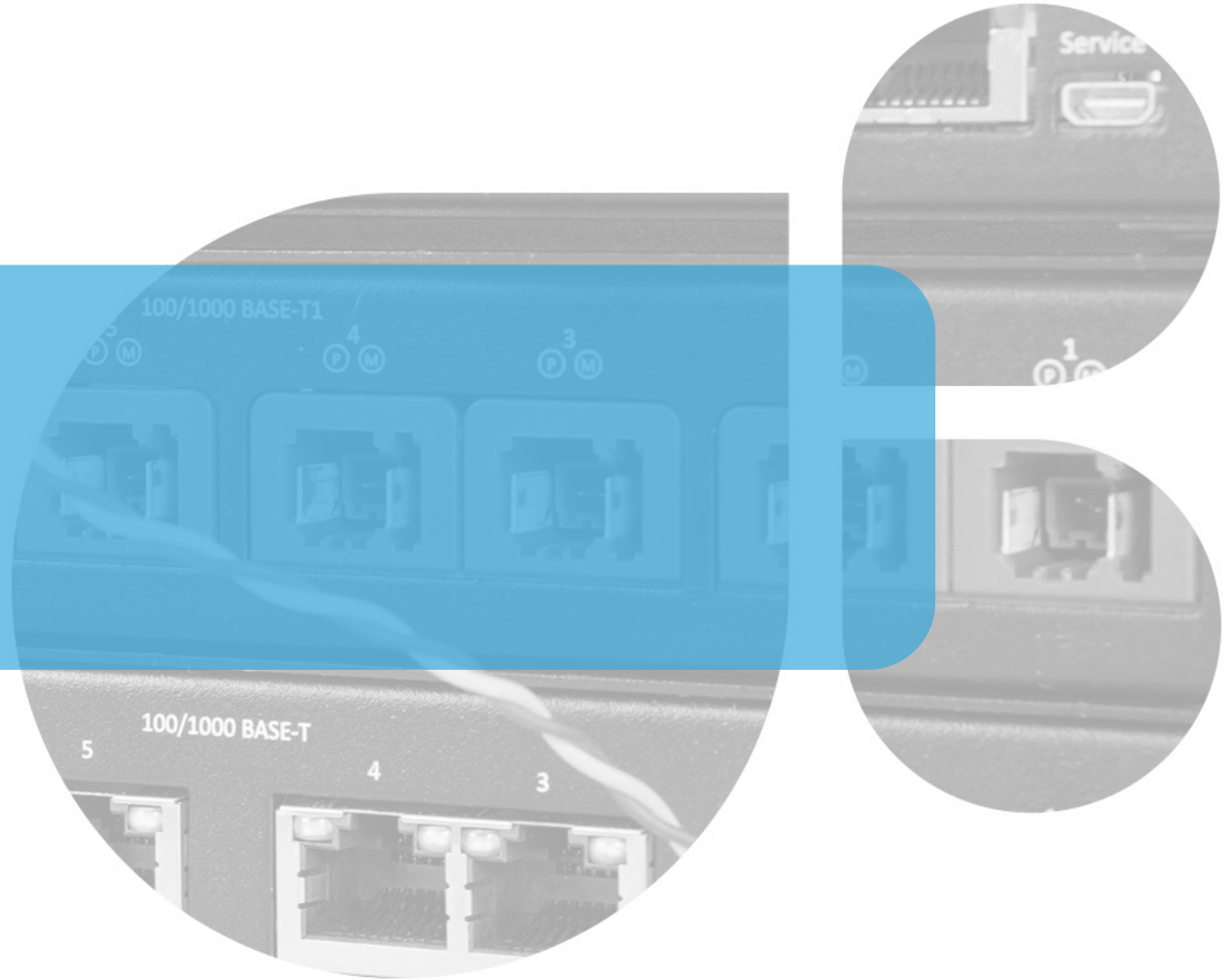
SECURITY PROTOCOLS AND APPLICATIONS

BEST FRIENDS OR WORST ENEMIES?

Antonio Gallego, José Galve, Dr. Lars Völker | Technica Engineering GmbH

SECURITY PROTOCOLS AND APPLICATIONS

TABLE OF CONTENT



SECURITY PROTOCOLS AND APPLICATIONS

#1 | MOTIVATION

MOTIVATION
PROBLEM SCOPE

Security is essential for the automotive development.

- Safety is only dependable, when the right Security is present.
- Regulations, like UN ECE R155, require Security to be considered.

Usability?

- Does Security slow down the development process?
- How transparently can Security be integrated?
- Application communication vs. Security?

We will focus on Network Security.

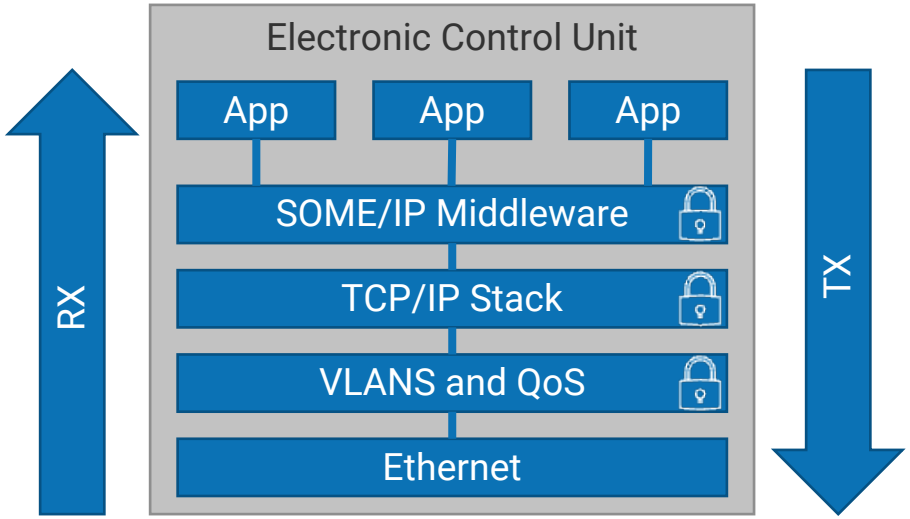
MOTIVATION
MODEL

Structure of ECU

- The SW architecture of an ECU is “somewhat” layered.
  - Layering mainly for data but not for control present.
  - Optimization: remove or combine layers.

Important Aspects

- Stack signals to applications when to communicate.
  - Most common: Ethernet Link up / Interface comes up.
  - Also: SOME/IP-SD, connections ready, etc.
- Security may be integrated into different layers.



SECURITY PROTOCOLS AND APPLICATIONS

#2| SECURITY PROTOCOLS EXAMINED

SECURITY PROTOCOLS EXAMINED

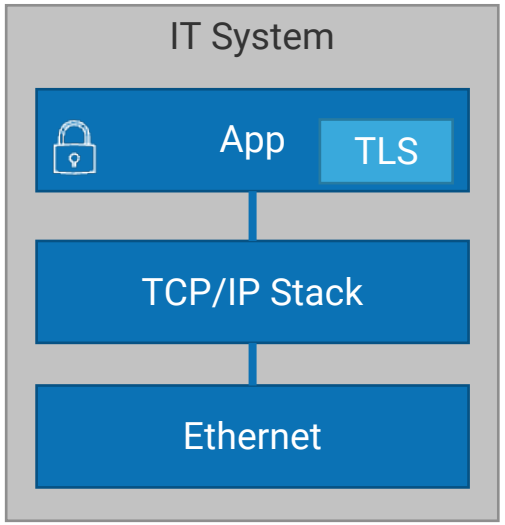
TLS/DTLS IN GENERAL (NON-AUTOMOTIVE)

- TLS is “typically” implemented as an application library.
  - Instead of sockets, you get secured sockets (Secure Socket Layer, SSL).
  - First usage: Webserver and browser.

- Application is fully aware of TLS or DTLS and controls it.
  - Typically, by preferences/config or for ”https” per URL.
  - You may bind the server application only to “secure sockets”.

- However, a compromised application can still communicate unsecure.
  - Firewalling and IDS/IPS try to cope with that.

- Designed for Internet and not so much for local network.



SECURITY PROTOCOLS EXAMINED

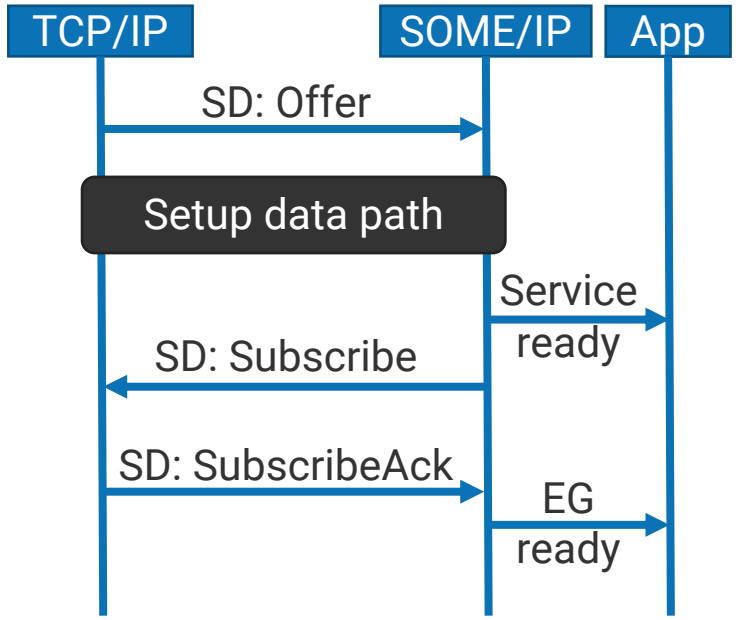
TLS/DTLS IN AUTOMOTIVE

SOME/IP Middleware abstracts stack/complexity

- SOME/IP sets up data path before telling App to start:
  - For UDP: no explicit setup required.
  - For TCP: Client opens TCP connection to Server first.

- What happens, when you add TLS/DTLS?
  - Adding TLS slows down the establishment of TCP connections.
  - However, adding DTLS changes a lot:
    - DTLS needs to secure the “connection” first, SOME/IP-SD does not wait.
    - Applications start to send unsecured data into DTLS handshake…
    - DLTS needs to be handled like TCP and not UDP (stacks missed this)!

- Lesson Learned: It is non-trivial to make Security transparent!



Simplified SOME/IP Client Flow

SECURITY PROTOCOLS EXAMINED

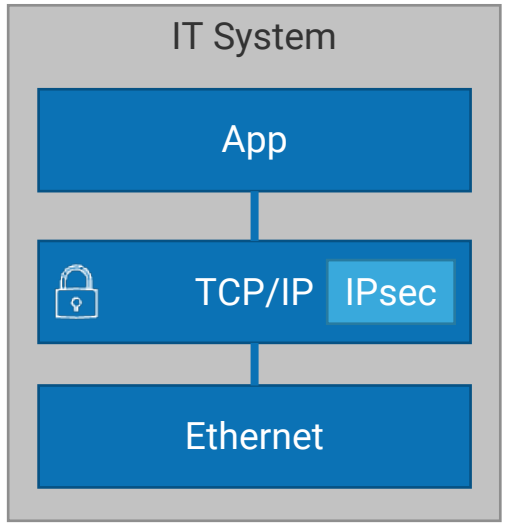
IPSEC IN GENERAL (NON-AUTOMOTIVE)

- IPsec is often used for VPNs.
  - Typically, part of the stack/operating system.
  - Often as “secure tunneling” of IP communication.

- IPsec matches traffic (like firewalling) and ”protects” it.
  - Example: All traffic to Corporate Headquarters go via VPN.

- Applications are (typically) not aware of IPsec.
  - The goal is to hide it from the user to ease usage.
  - Firewalling and network design stop communication, when VPN tunnel stops.

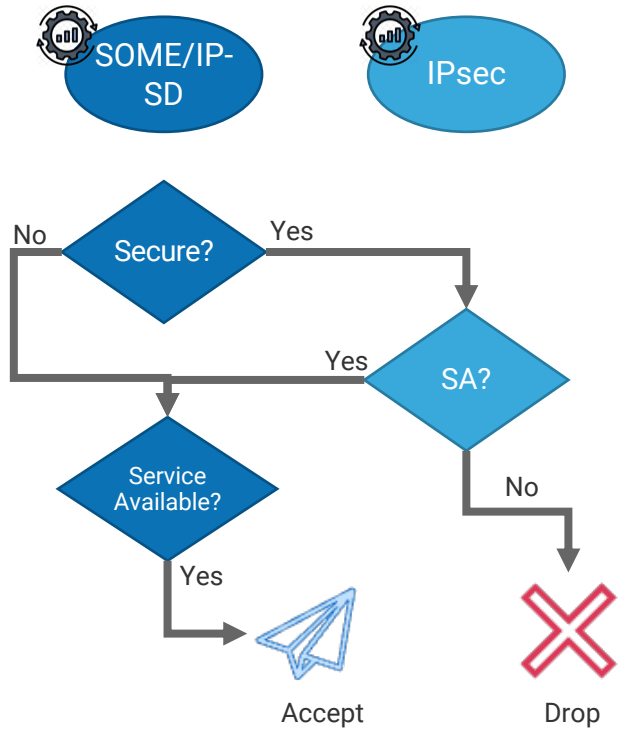
- Designed for Internet and not so much for local network.



SECURITY PROTOCOLS EXAMINED
IPSEC AND APPLICATIONS

IPsec and Firewalling hide inside TCP/IP stack

- IPsec protection is based on Security Policy (~firewalling).
  - For example: protect traffic to Destination IP 1.2.3.4 and port 12345/udp.
- SOME/IP stacks and IPsec start in parallel. Peers might be late.
  - What can go wrong?
- Surprise: SOME/IP and IPsec need to communicate.
  - Does this communication require IPsec? Ready to do so?
  - IPsec standard does not really explain this.
  - SOME/IP Endpoints may determine which traffic must be protected, i.e., based on port ranges.
- Lesson Learned: Security standard does not discuss these issues.



SECURITY PROTOCOLS EXAMINED

LESSON LEARNED.

What did we learn so far?

- Security protocols (like TLS/DTLS, IPsec) are not as transparent as expected.
- When ignoring that, things can really go wrong.

Discussion:

- Start to communicate as early as possible?
- Start to communicate after Security is ready?
- When using SOME/IP Middleware, these issues and complexity can be hidden.
- What happens with communication besides SOME/IP (e.g., NM, DoIP, …)?

SECURITY PROTOCOLS AND APPLICATIONS

#3| HOW CAN WE IMPROVE?

HOW CAN WE IMPROVE?
HELP THE APPLICATION DEVELOPERS

Divide and Conquer.

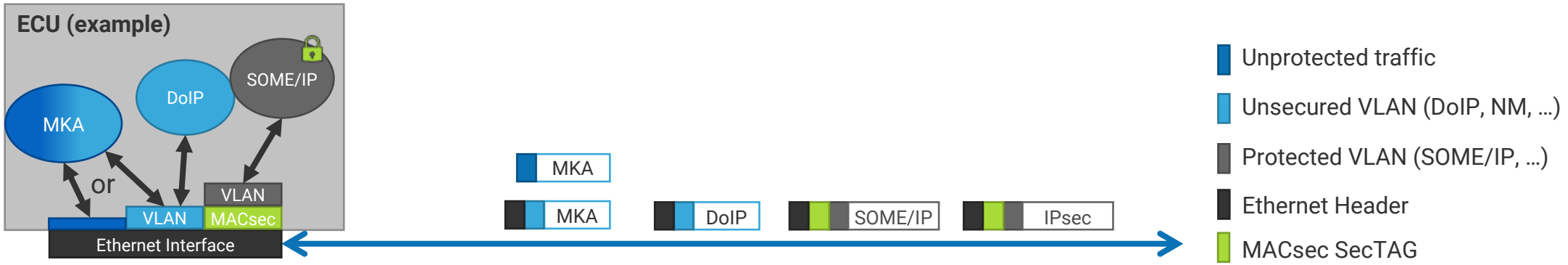
- Most application developers are no experts in secure communication.
  - Goal: hide the security without reducing the security.
  - Let the application developers focus on their work.

- Strategy:
  - Do not create special security APIs towards applications!
  - Let the integrator and security/communication developers worry about security.
  - Align security and communication with standardized APIs.
  - Create solutions which are capable of updates to foster innovation.
    - Secure the platform and not only individual use cases.

HOW CAN WE IMPROVE?
EXAMPLE: MACSEC

MACsec as the next State-of-the-Art Automotive Network Security solution

- Aligning “MACsec ready” with “Ethernet link up”.
  - On linkup of physical interface, run MKA but do not signal application.
    - Link up signaled to applications after MKA signals ”MACsec ready”.
  - MACsec configuration via “key installation/diagnostics”.
    - Issue for applications? Diagnostics?
- Solution: Use virtual interfaces by creating “unprotected VLANs”.



HOW CAN WE IMPROVE?
EXAMPLE: MACSEC (2)

MACsec as the next State-of-the-Art Automotive Network Security solution

- Low startup performance may require “unsecure communication”
- Solution: Make MACsec so fast that this is not necessary.
  - With shared symmetric secret (CAK), we achieved ~14ms[1] startup or faster.

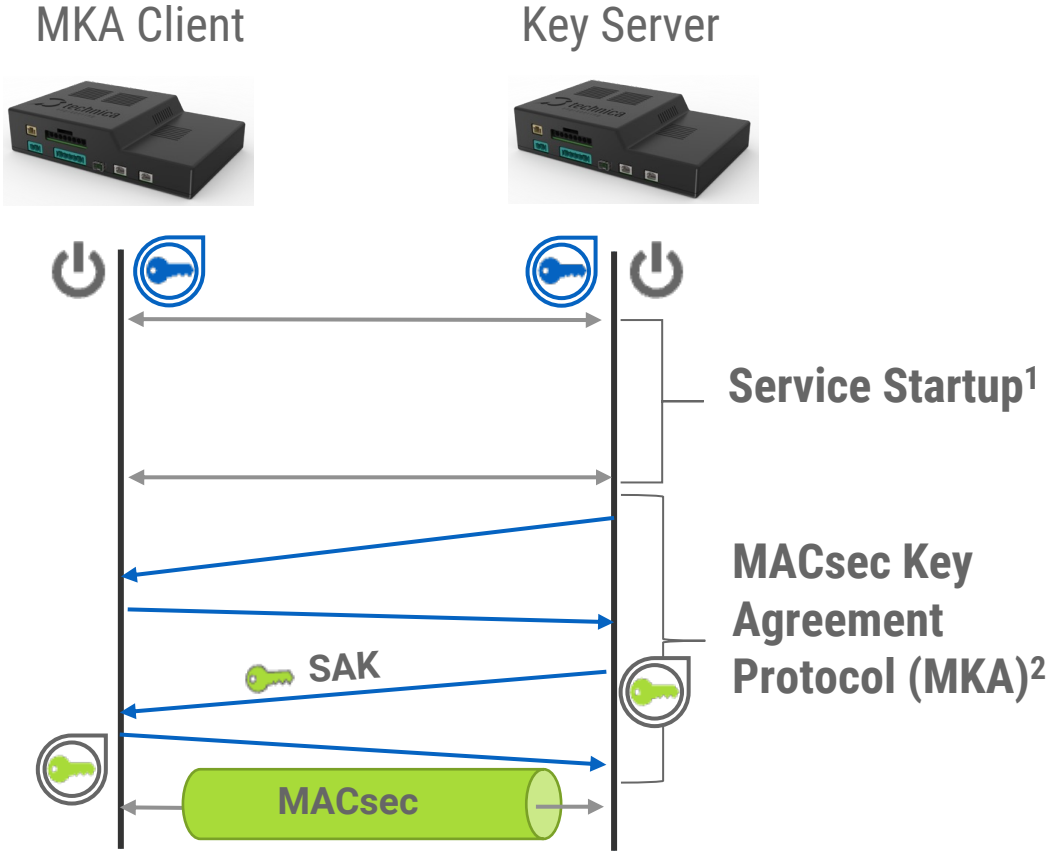
Technica Development Boards

Technica Capture Module 1000 High
- Logging in between both peers with HW Timestamp

[1]using an exemplary external PHY, for other semiconductors this may be different. Detailed information can be found in 2021-06-22_VDI_CyberSecurityVehicles-DrLarsVoelker_v.1.0a.pdf

HOW CAN WE IMPROVE?

EXAMPLE: AUTOMOTIVE MACSEC



Results:

Type	Startup[1]	MKA[2]
IT solution (Open Source)	~2000ms	~3000ms
Automotive solution	3-4ms	13-17ms

[1]**Startup**: Power-in until whenever the port is reachable and sends its first online message.
[2]**MKA:** first MKA message, until the SAKs are installed and the first MACsec Frame is sent.

→ **Automotive MACsec starts so fast that all applications can wait for it!**

SECURITY PROTOCOLS AND APPLICATIONS

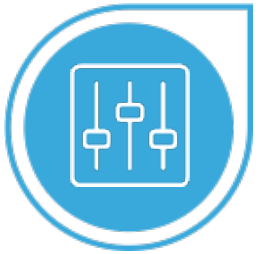
#4| CONCLUSION

CONCLUSION
LESSON LEARNED

What did we learn?

- Integrating Security can be very challenging.
- Holistic understanding of stack is essential!
- Protecting the ”local network” in a car is non-trivial with TLS/DTLS and IPsec.

What to do?



- Hide the security from applications. MACsec can help with this.
- Avoid a special “Security API” towards the application.
- MACsec can help you create the secure platform to update later with ease.

If done right, applications and security protocols are best friends again…

CONCLUSION
ONE MORE THING…

First Automotive MKA daemon goes Open Source!

Where? How? When?

- Technica-Engineering/MKA.SW.Module · GitHub
- GPLv2 licensed.
- Commercial license available.
- Late November 2022.

What is supported?

- MKA tuned for Automotive Networks.
- Standard APIs compatible with security suites (OpenSSL, WolfSSL, …).
- Available for Linux based OS.

Get in touch for more information and updates!

TECHNICA ENGINEERING GmbH

Leopoldstraße 236
D - 80807 München

ANTONIO GALLEGO
Group Leader Security
antonio.gallego@technica-engineering.de
+49 (0) 176 207 42953