PnS for CAN

Plug-and-Secure Communication for CAN

Vector CAN FD Symposium
Stuttgart, Germany
February 16th 2017

Dr. Arthur Mutter

BOSCH

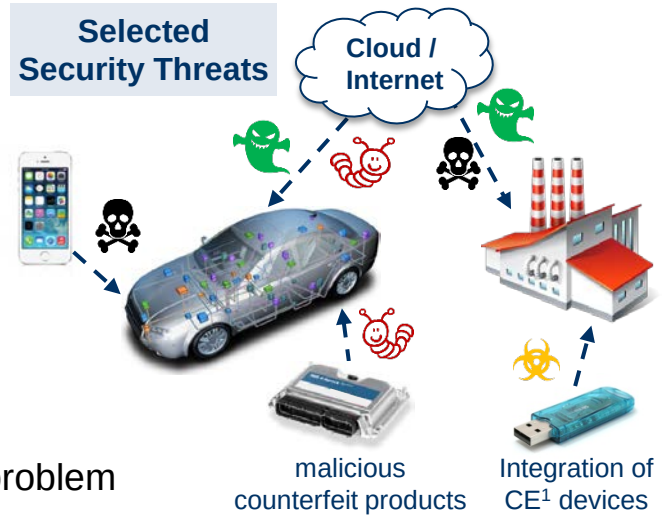
2015: 72 million new cars sold

Source: OICA via Statista.de

➔ Huge Potential Damage

Especially via Remote Attacks!

Motivation



Facts

- Current trends (e.g. Cloud/Internet connectivity) lead to novel & serious security threats
- Today's CAN networks are often hardly secured
- Cryptographic methods may help (e.g. message auth.)

However

- Key agreement and distribution is not a solved or trivial problem
  Reasons: security, effort, computational complexity, price
- Keys have not been attacked – simply because they did not exist

Our Idea: Plug-and-Secure

A novel approach for completely automated & secure key establishment of very low complexity for CAN networks ("plug-and-secure")

- **Especially suitable against software-based & remote attack scenarios**
- **Basic Idea: Exploit special properties of CAN bus (dominant / recessive bits)**

[1]Consumer Electronics

Agenda

- Basic idea
- Major benefits
- Security considerations
- Implementation options and details
- Demonstrator

[1]Consumer Electronics

Basic Idea

1

Random Number Alice

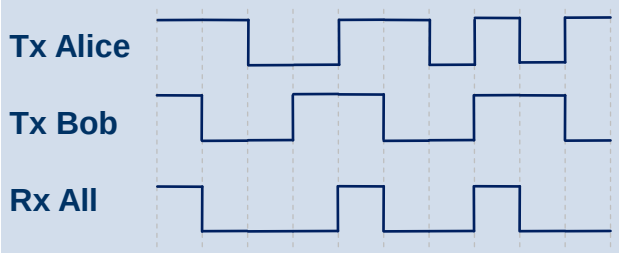
1 1 0 0 1 1 0 1 0 1

Random Number Bob

1 0 0 1 1 0 0 1 1 0

2

Overlay of signals on the CAN bus



3

Shared Secret = Tx Alice + Rx All

1 0 1 0 1

Eve cannot calculate shared secret

Inverse Shared Secret = Tx Bob + Rx All

0 1 0 1 0

BOSCH

Details

1a Generate a random bit string of length N

S_{Alice} = 1 1 0 0 1 1 0 1 0 1

Alice	Bob	Bus
0	0	0
0	1	0
1	0	0
1	1	1

CAN Bus = Alice AND Bob

Generate a random bit string of length N

S_{Bob} = 1 0 0 1 1 0 0 1 1 0

1b Replace: 0 → 01, 1 → 10

S_{Alice}^* = 10 10 01 01 10 10 01 10 01 10

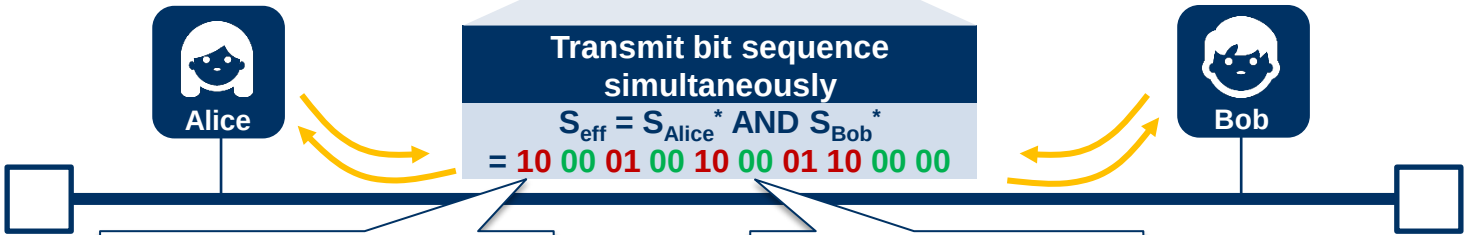
Replace: 0 → 01, 1 → 10

S_{Bob}^* = 10 01 01 10 10 01 01 10 10 01

2 Transmit bit sequence simultaneously

$S_{eff} = S_{Alice}^* \text{ AND } S_{Bob}^*$

= 10 00 01 00 10 00 01 10 00 00



3 Discard bits corresponding to '01 or '10' in S_{eff} in initial bit sequences S_{Alice} / S_{Bob}

S_{Alice} = X 1 X 0 X 1 XX 0 1

1 0 1 0 1

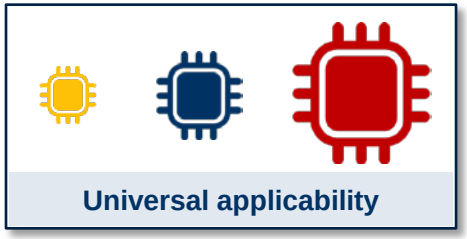
Inverse sequences = shared secret

S_{Bob} = X 0 X 1 X 0 XX 1 0

0 1 0 1 0

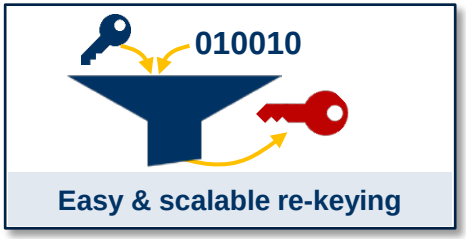
BOSCH

Major Benefits



Universal applicability

Low bandwidth requirements



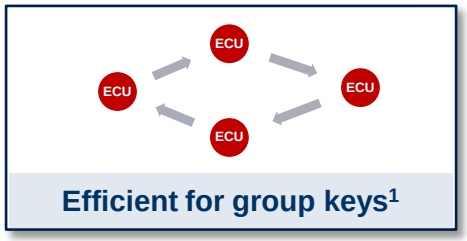
Easy & scalable re-keying

Low complexity & low cost

Plug-and-Secure Communication for CAN

Works w/ any CAN controller

Simplicity / Ease-of-Use



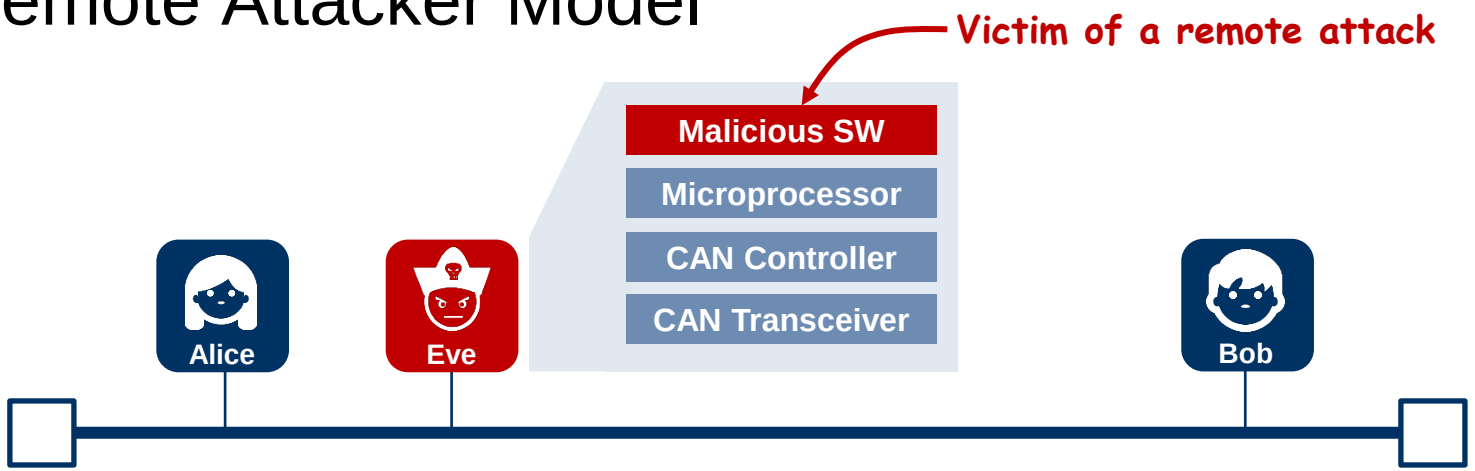
Efficient for group keys[1]

Seamless integration into CAN ecosystem

[1]Technical paper: http://eprint.iacr.org/2016/601

BOSCH

Remote Attacker Model



Assumptions

1. Eve is using standard HW with modified (malicious) SW
2. Eve may eavesdrop on all messages exchanged on the CAN bus
3. Eve may inject arbitrary bits on the CAN bus (via the CAN transceiver)

→ **Highly relevant attacker model due to easy scalability of attacks!**

Remote Attacks

Idea: Passively eavesdrop on the channel during key setup

- Fact: Alice + Bob derive secret only from secure bit pairs (“00” on bus)

➔ A passive Eve cannot determine the established secret bits

Idea: Actively interfere with key establishment procedure

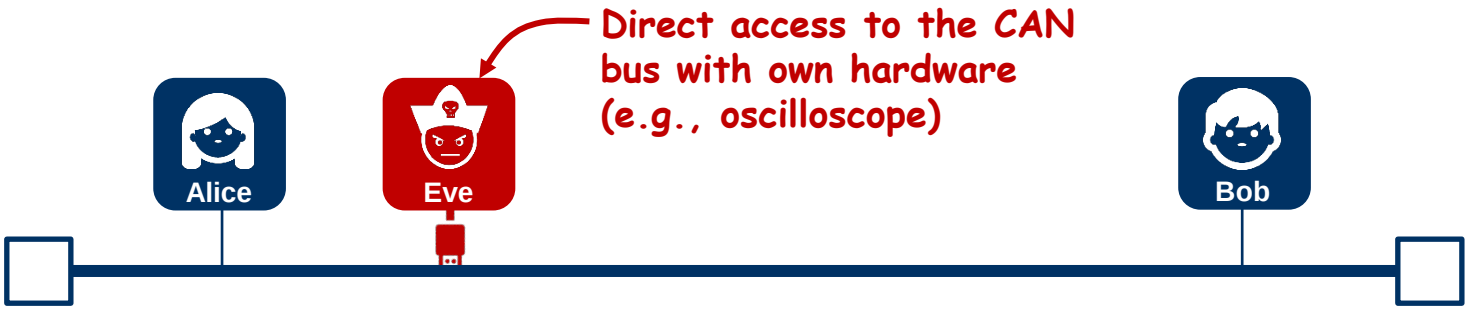
- Action: Eve transmits dominant bits ➔ leads to “00” bit pairs on bus
- Result: Alice + Bob derive different secrets
- Solution: Perform key verification after key generation

➔ An active Eve can prevent a successful key establishment

➔ An active Eve cannot determine or influence the established keys

BOSCH

Attacker Model with Physical Access to CAN Bus



Principle Threats

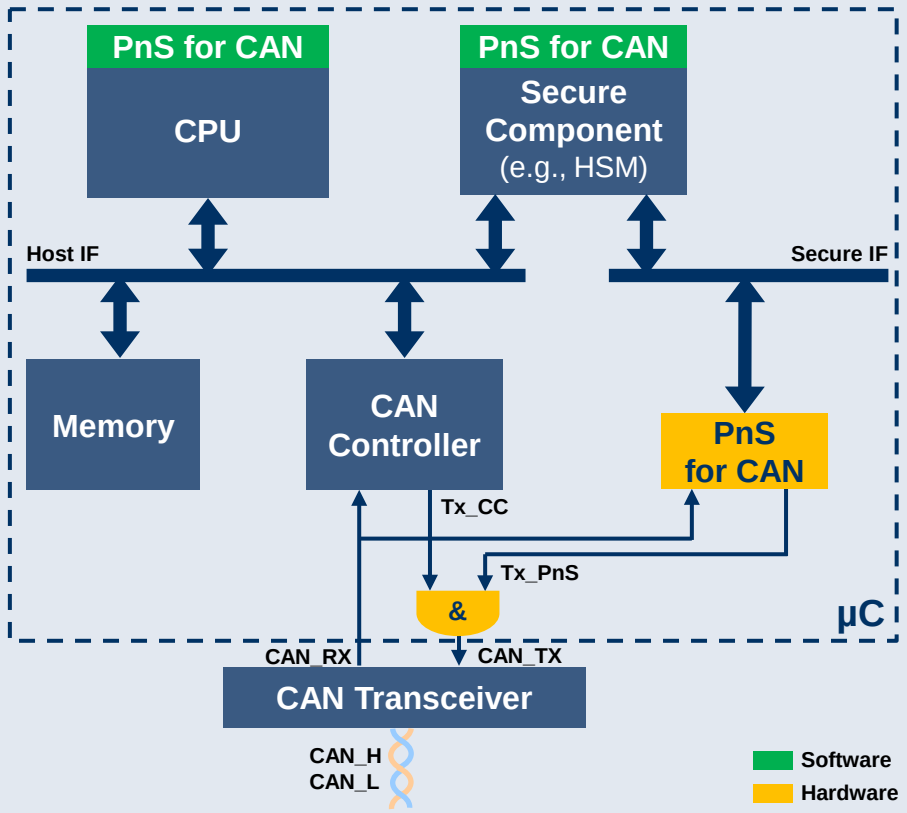
- Physical access enables more sophisticated attacks (e.g., exploitation of timing or attenuation effects)
- Attacker needs detailed knowledge of the CAN bus

BUT:

- With physical access, an attacker could compromise a vehicle much easier (e.g., cut a cable)
- Attacks requiring physical access do not scale; threat with physical access always existed
- Countermeasures are possible → e.g., artificial (random) jitter in bit timing

Implementation: PnS Module in Microcontroller

in Microcontroller



Properties

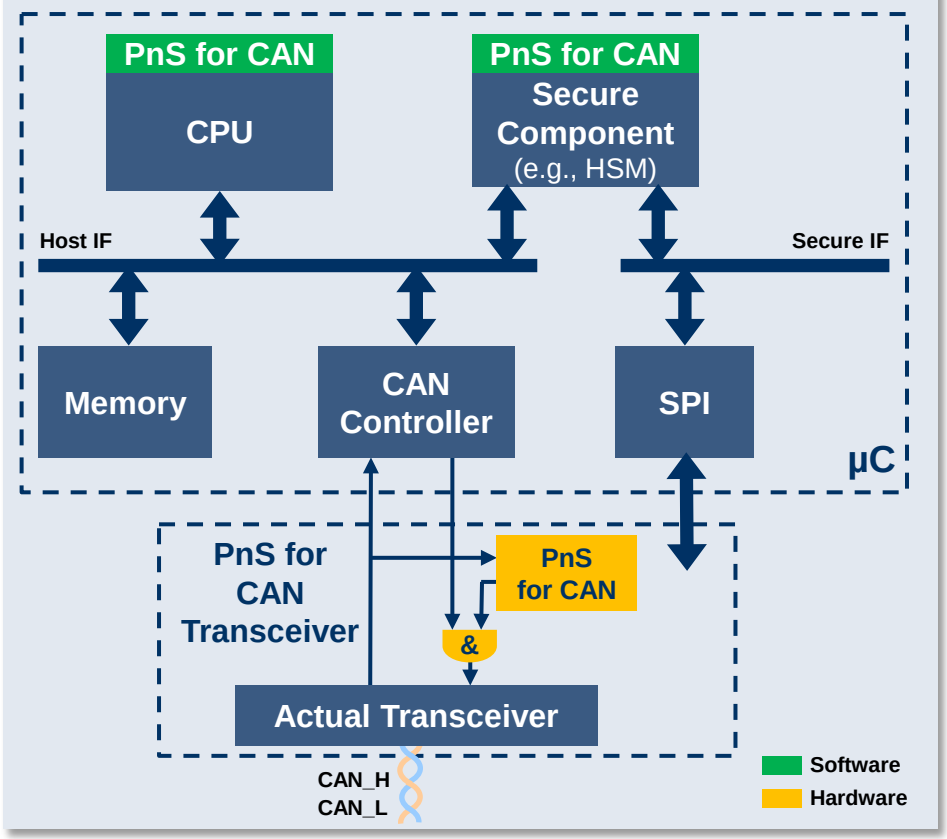
- "PnS for CAN" module
  - is a reduced CAN controller → low costs
  - connects to the CAN bus in parallel to CAN controller
  - competes with the on-chip CAN controller and other CAN devices via arbitration
  - is compatible to any CAN controller
- Separation of core functions in dedicated HW module is good from a security point of view
- Secure component (optional) stores keys and performs crypto functions[1]

[1]If no secure component is available, "PnS for CAN" module might be directly connected to CPU

BOSCH

Implementation: PnS Module in CAN Transceiver

in CAN Transceiver



Properties

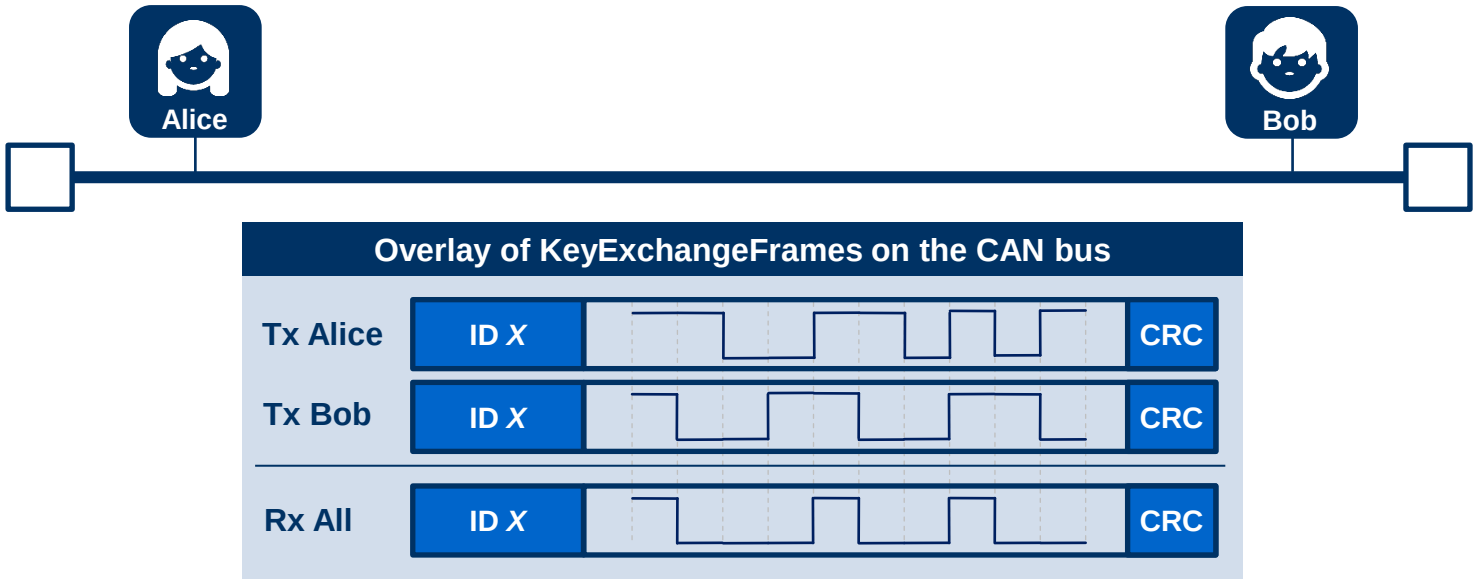
- No modifications of existing µC HW necessary
  → quick upgrade path
- May be combined with any existing µC
- Communication between “PnS for CAN” module in TRX and µC via SPI
- Encapsulation of core functions in HW module good from a security point of view
- Secure component (optional) stores keys and performs crypto functions[1]

[1]If no secure component is available, “PnS for CAN” module might be directly connected to CPU

BOSCH

Synchronization of Frame Transmission (I)

- Target: KeyExchangeFrames of Alice and Bob need to overlay

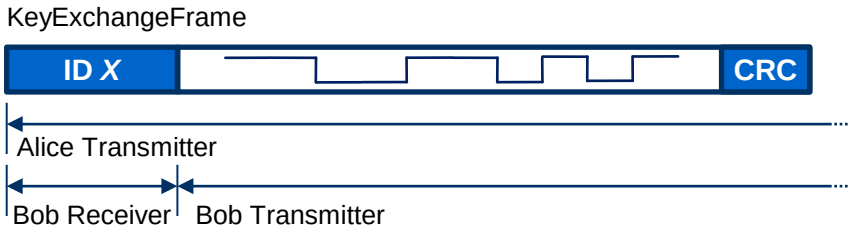


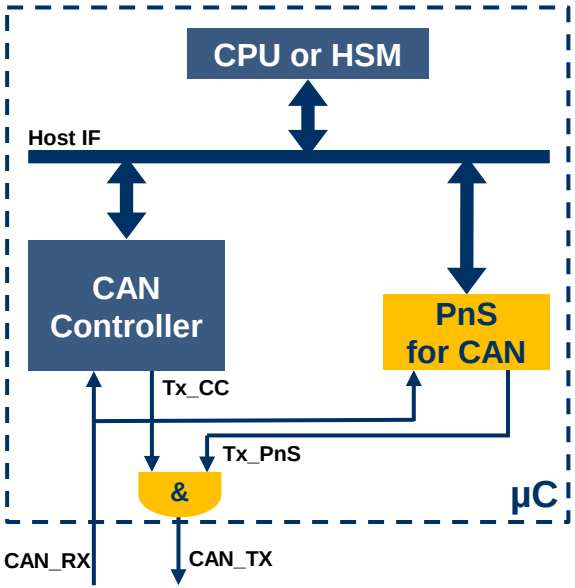
- Challenge: Alice and Bob need to synchronize
- Therefore: One node triggers the other node
  E.g.: Alice tells Bob to start TX of KeyExchangeFrame

Synchronization of Frame Transmission (II)

Procedure

- Alice/Bob: CPU configures frame ID *X* in PnS
- Alice: Sends a frame with ID *X*
- Bob: When PnS detects frame ID *X* on CAN bus, it switches its status from Receiver to Transmitter



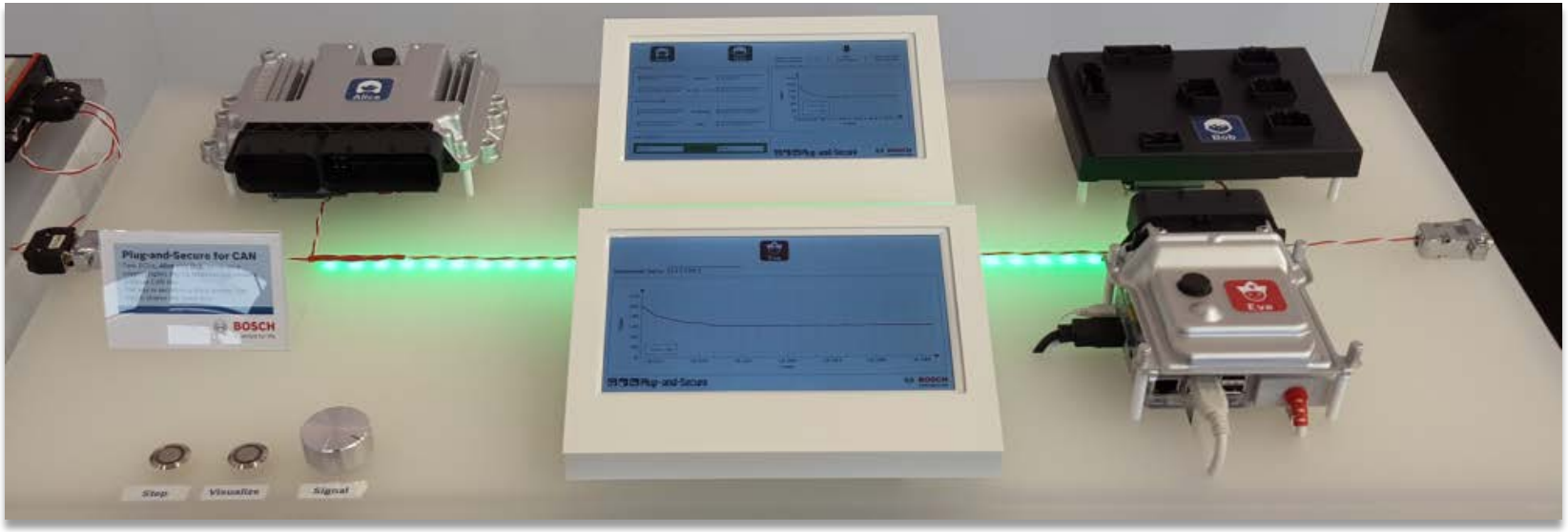


Properties

Independent, precise, simple

Demonstrator

- Feb. 2016: “Embedded World” ➔ Demo
- Nov. 2016: “Electronica 2016” ➔ Demo



Summary

- **Task**
  Secure Key Establishment

- **Properties**
  Very low complexity, high efficiency, low cost

- **Operation**
  On any CAN bus (Classical CAN or CAN FD)

- **Implementation**
  PnS Module required in Microcontroller or Transceiver

PnS for CAN

- **Major Strengths**
  Remote / SW-based attacks, Automated key exchange

BOSCH