HoliSec

Holistic Approach to Improve Data Security

AUTOSAR Secure Onboard Communication (SecOC) Introduction to SecOC and a proposal for key management methodologies

Presenter: Bashar Dawood

March 26, 2019

VINNOVA FFI VOLVO VOLVO ARC CORE CHALMERS ASSURED SECURITY CONSULTANTS RI.SE Research Institutes of Sweden

Outline

- Why is secure on-board communication necessary?
- What is AUTOSAR SecOC?
- SecOC Basics
- SecOC opportunities for standardization
- Key management methodologies for symmetric key authentication
- Key management methodologies for asymmetric key authentication

Why is secure on-board communication necessary?

- Mechanical to electrical control of vehicles.
- More functionallity and flexibility but,
- Exposes safety-critical functions to secuirty threats.
- Modern vehicles contain hundereds of interconnected ECUs
- Controlling numerous domains e.g. Steering, braking, etc...
- Unauthorised bus access poses multiple, possibly life threatning risks

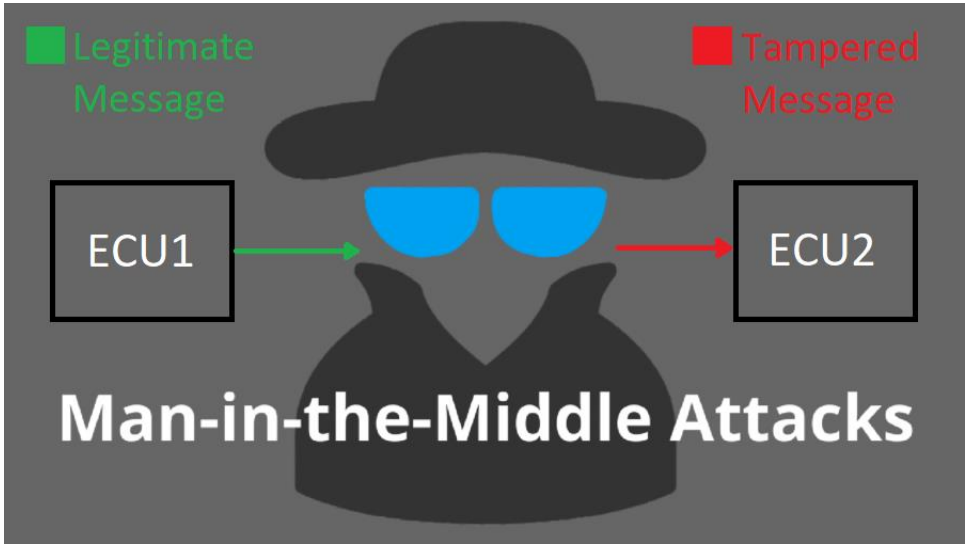
Why is secure on-board communication necessary?

Possible threats:

- Replay Attacks
- Tampering and man-in-the-middle attacks
- Random Errors

Possible reprecutions:

- Access to physical controls
  - Compromised safety of passengers
- Tampering with mileage, emissions, etc
  - not meeting regulations
- Intellectual property theft



AUTOSAR SecOC

- An AUTOSAR BSW Module
- Parallel to the PDU Router BSW
- Efficient authentication of critical data
- Protection on the level of Protocol Data Units (PDUs) – Network Agnostic
  - CAN✓
  - Ethernet✓
  - Flexray✓
  - etc....✓
- Appends PDUs with authentication information

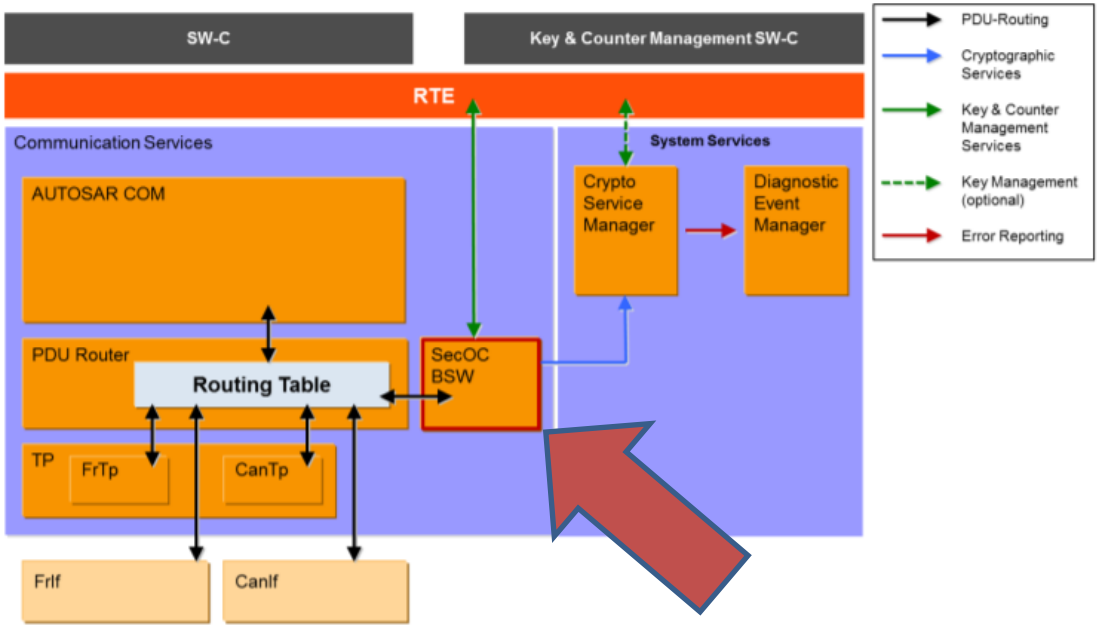
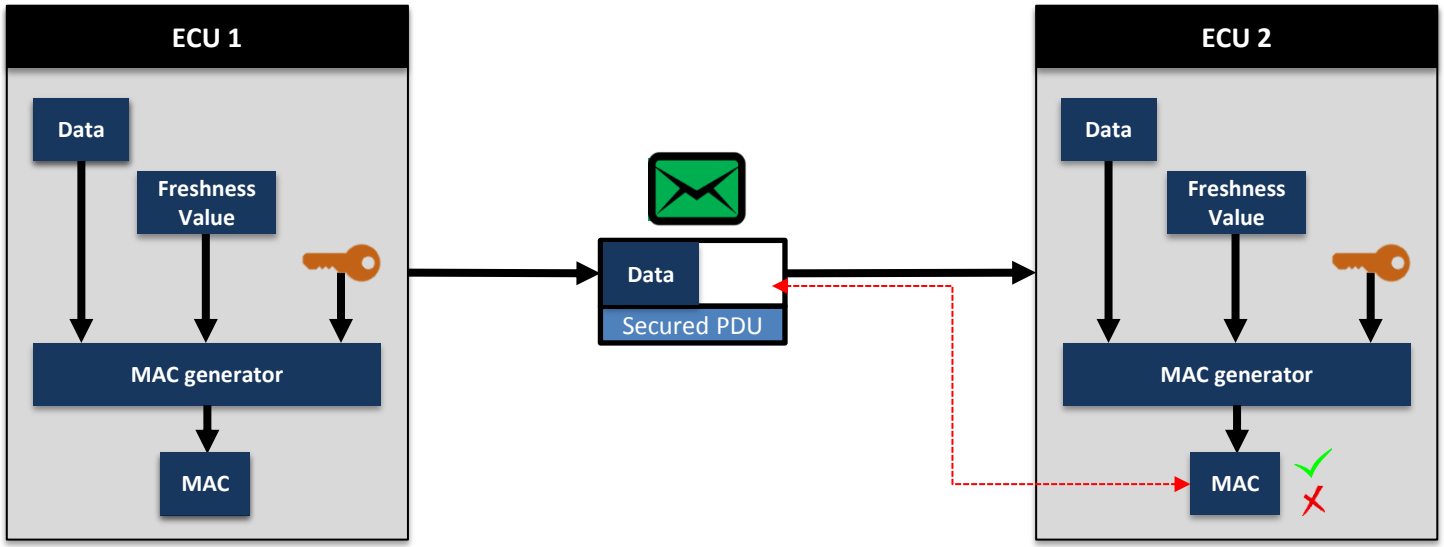


Figure 1: Integration of the SecOC BSW

SecOC Mechanism



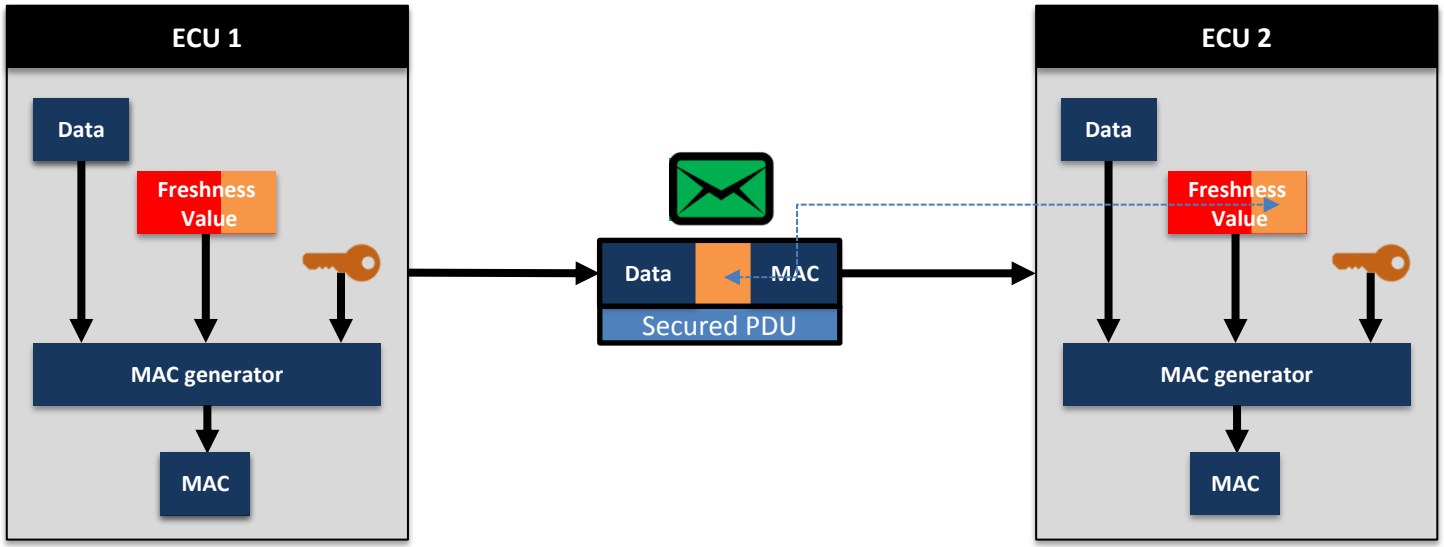
SecOC does **NOT** protect against eavesdropping!

SecOC opportunities for standardization

Freshness Value

- Added variable for extra secuity
- Prevents replay attacks, synchronized freshness values across all ECUs
- Incremented per message sent
- Appended to PDU pre-MAC generation
- Managed by Freshness Value Manager (FVM)
- Multiple configurations
  - No Freshness Value
  - Freshness Value based on Single Freshness Counter
  - Freshness Value based on Single Freshness Timestamp
  - Freshness Value based on Multiple Freshness Counters
  - Truncated Freshness Value in message packet
    - More efficient synchronization
- FVM implementation in application layer, OEM specific – Christian Sandberg

SecOC Mechanism – Truncated Freshness Value



Key Management

- Symmetric Key Authentication via Secret Keys
  - Same key stored on all interconnected ECUs
  - Primary Authentication mechanism for SecOC -> generate MACs
- Asymmetric Key Authentication via Private/Public Keypair
  - Each ECU has different Private/Public Keypair
  - Supported with SecOC with modifications -> generate signatures
- SecOC utilises Crypto Service Manager (CSM) Module
  - Secure key access & storage via Crypto Interface(CRYIF)
  - MAC/signature generation
- Currently OEM-Specific key management methodologies

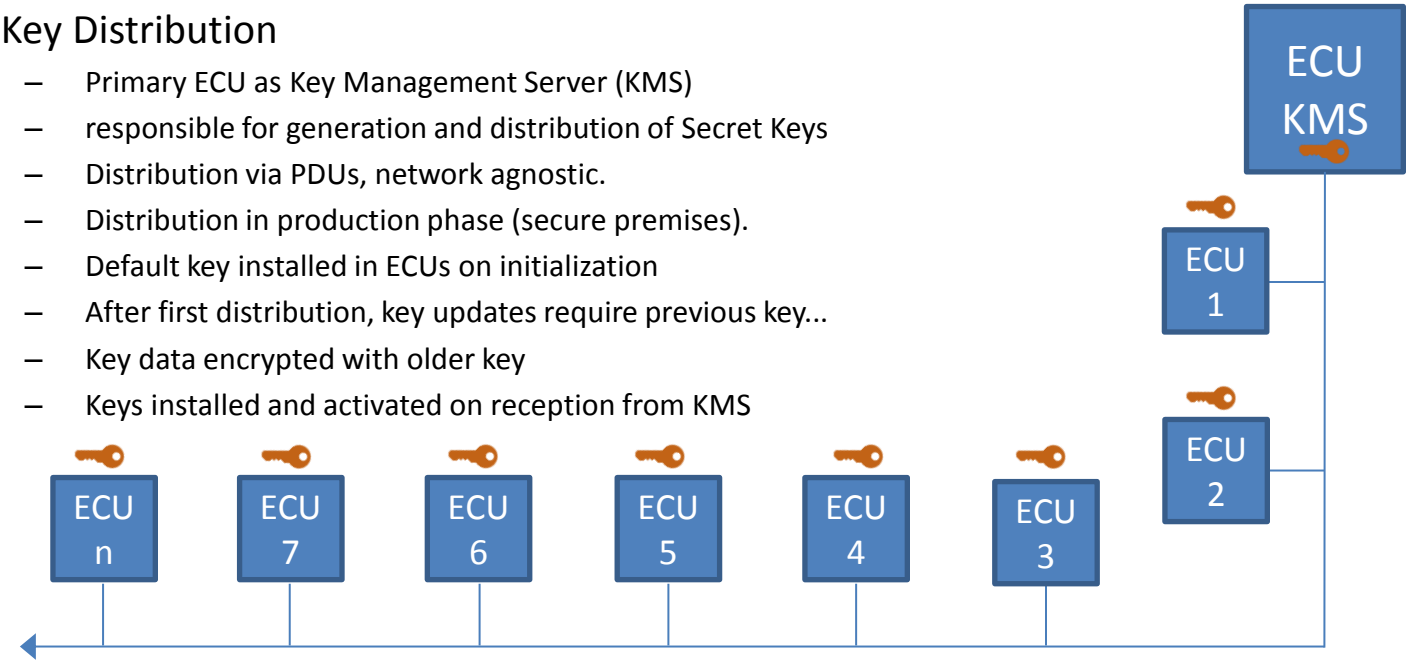
Symmetric Key Management

Symmetric Secret Key Management

- Key Generation
  - Random keys generated with approved Random Bit Generators (RBG)
  - Using approved cryptographic modules (ECU hardware)
  - Country specific compliance standards
  - Europe: chapter 2 in “Algorithms- key size and parameters”
    European Union Agency for Network and Information Security (ENISA)
  - USA: FIPS 1402 compliancy National Institute of Standards and Technology (NIST) standards
  - **AUTOSAR CryIF module must be implemented to international standards**

Symmetric Secret Key Management

- Key Distribution
  - Primary ECU as Key Management Server (KMS)
  - responsible for generation and distribution of Secret Keys
  - Distribution via PDUs, network agnostic.
  - Distribution in production phase (secure premises).
  - Default key installed in ECUs on initialization
  - After first distribution, key updates require previous key...
  - Key data encrypted with older key
  - Keys installed and activated on reception from KMS



Symmetric Secret Key Management

- Key Storage and Access
  - Keys are stored in and accessed from Secure On-Chip Hardware
  - Different Implementation for different hardware:
    - Arm TrustZone®
    - NXP Secure Kinetis®
  - No access to secured flash (where keys are stored) via debug port
  - Accessed via AUTOSAR SecOC Module -> CSM Module -> CryIF Module (software layer)
  - CryIF Module -> MCAL (hardware layer)
  - **No use of Software Crypto, unsecured**

Symmetric Secret Key Management

- Compromised Keys
  - Probabilistic tamper detection decision from ECU (not covered here)
    - Drastic behavioural changes
    - Clocking changes
    - Consecutive false replays
  - Vehicle ceases to function (if parked), engine off
  - Tamper event information to be stored in NVM
  - On-board communication halted
  - Tamper event & keys extracted from ECU and documented by trusted party (OEM service shop)
  - New keys injected to ECU by trusted party (secret procedure)
  - Applies to individual ECUs (communication busses can't be trusted)
  - Costly but necessary

Symmetric Secret Key Management

- Archiving and Destroying old keys
  - Keys have a lifespan of 30 days
    - *KMS distributes new keys to all ECUs*
  - Three keys to be stored in secure flash:
    - *Latest key and previous two generations.*
  - No need to store older generation keys (waste of space)
  - Previous generation keys used to recover corrupted keys.

- Recovering corrupted keys (corrupted flash)
  - Distress message broadcast (after fail key detected)
  - Fall back to previous archived key
  - If still corrupted, one last try...
  - Else, treat as compromised.
  - On recovery, KMS updates **all** ECUS with fresh keys.

Asymmetric Keypair Management

Asymmetric Keypair Management

- Keypair consists Public & Private keypairs
- Supported by SecOC with modifications
- Sender uses its private key to generate signature (similar to MAC)
- Recievers use sender's public key to verify signature
- Inefficient, but more secure
- **No support of truncated signature, unlike MAC**

Asymmetric Keypair Management

- Generating Keypairs
  - ECUs generate their own keypairs (via CSM)
  - KMS signs public keys (certificate authority)
  - Only signed public keys are active
  - *Default keypair per ECU type in production phase (presigned)*
- Distribution
  - KMS broadcasts each signed public key to ALL ECUs
  - other ECUs store public keys in whitelist
  - Keypair now active
- Revoking
  - KMS broadcasts illegitimate public key to ALL ECUs to blacklist
  - Affected ECU to regenerate new keypair and have signed by KMS

Thanks For Listening!