Secure Global Time Synchronization

Tarav Shah, Pavithra Kumaraswamy

8 December 2022 | AUTOSAR NA User Group

Agenda

Secure Global Time Synchronization

01

Introduction

- Motivation
- Standardization Bodies
- Use Cases

02

Integrated Security Mechanism

- Security mechanism on Ethernet
- Security mechanism on CAN
- Security mechanism on FlexRay

03

Security Mechanism - Ethernet

- Additional Considerations
  - Correction field protection
  - Pdelay protection

04

Architecture

- Software Architecture
  - Variants
  - Evaluation of variants

05

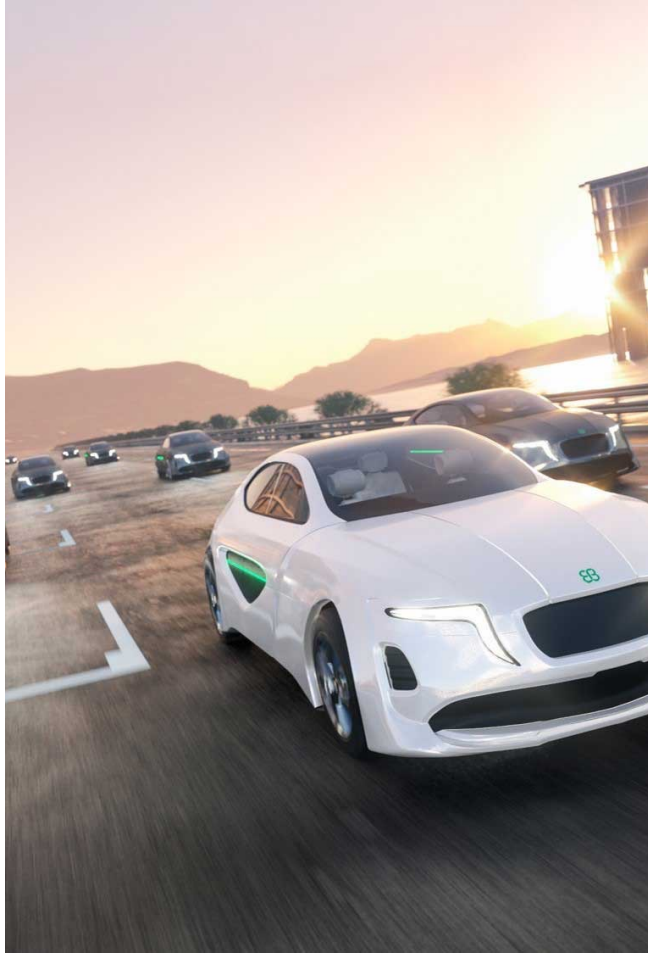
Other Security Mechanisms

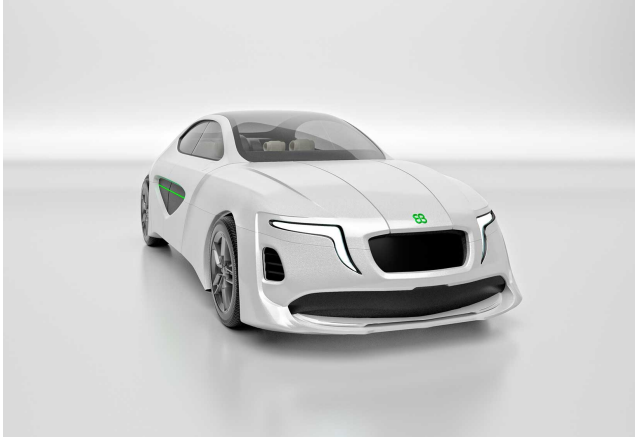
- External Security mechanism
- Architectural solution
- Monitoring and Management

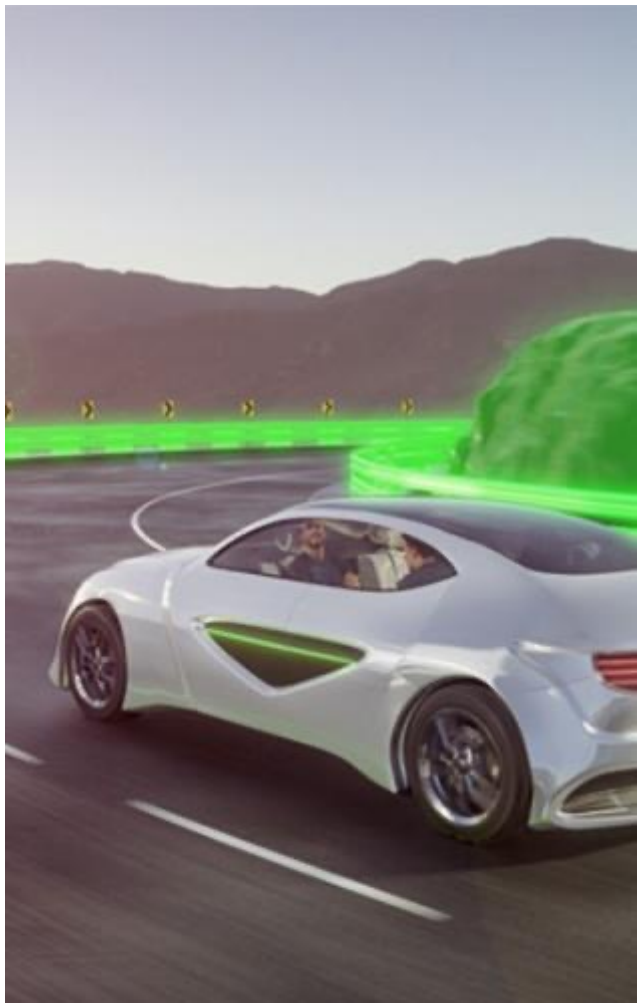
06

Challenges to solve

- Effective configuration
- Deployment of cost-effective security mechanism



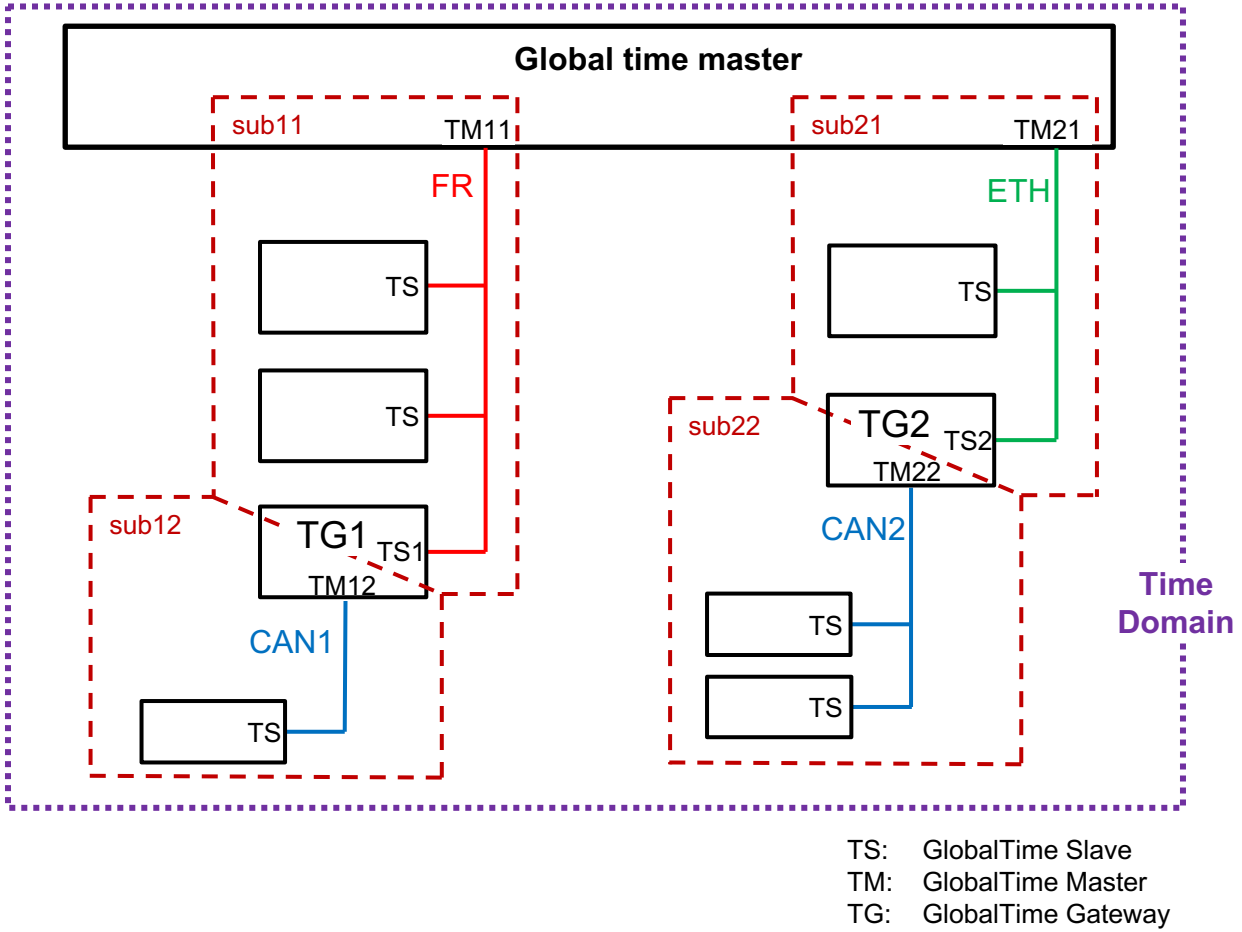




Introduction

Motivation – Secure Global Time Synchronization

- Use cases of Global Time Synchronization (GTS)
  - Synchronization of runnable entities
    - Synchronous sensor data read across ECUs
    - Synchronous actuator triggering across ECUs
  - Provision of absolute or relative time
    - Temporal correlation (event data recordings, data storage)
    - Time expiry monitoring (certificate-based authentication)
- Application of GTS is in safety-critical, time-critical and security-critical applications
- Issue with unsecure GTS
  - Potential security risk in vehicle due to
    - False time
    - Accuracy degradation
    - Denial of Service (DoS)



TS: GlobalTime Slave
TM: GlobalTime Master
TG: GlobalTime Gateway

Introduction

Standardization Bodies – Secure Global Time Synchronization

- gPTP [IEEE 802.1AS – 2011]
  - Security protocol not included
- AUTOSAR
  - R22-11 extends the GTS with security protocol [draft]
    - Concept responsibility from Elektrobit
      - Pavithra Kumaraswamy
      - Andrei Rus
    - Concept supported from WG-TSY
  - R23-11 extends the validation of security protocol of GTS [released]

- RFC 7384
  - Security requirement for time protocols [PTP, NTP]
- IEEE 1588
  - Annex P: (Informative) Security

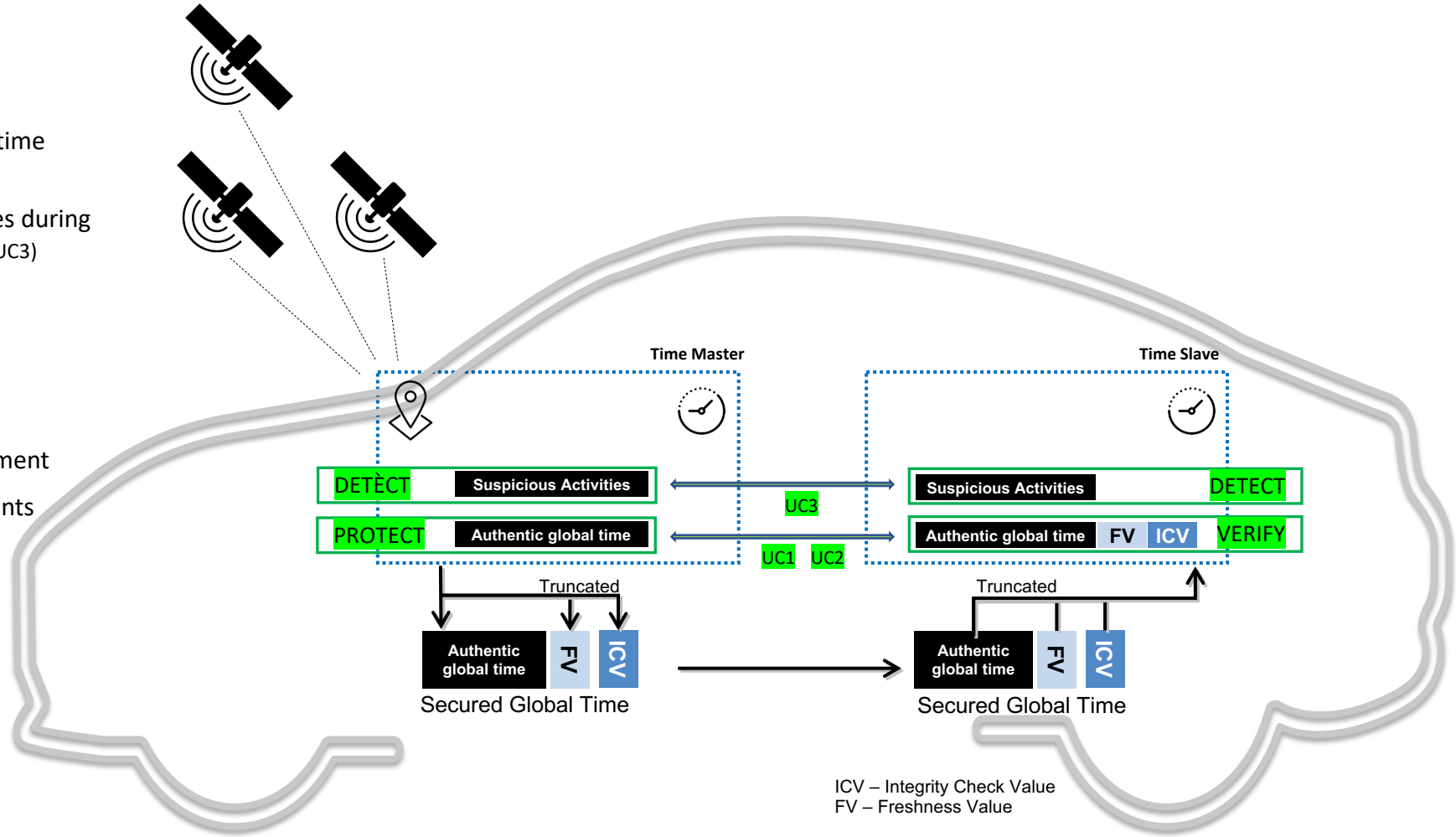
Introduction

Use Cases – Secure Global Time Synchronization

- Use Cases of SGTS
  - Protect and Verify the global time (UC1, UC2)
  - Detect the suspicious activities during global time synchronization (UC3)

- Dependent Use Cases
  - Cryptographic operation
  - Cryptographic credential management
  - Handling of security and error events
  - Freshness Value management

  [Covered as part of System Design]



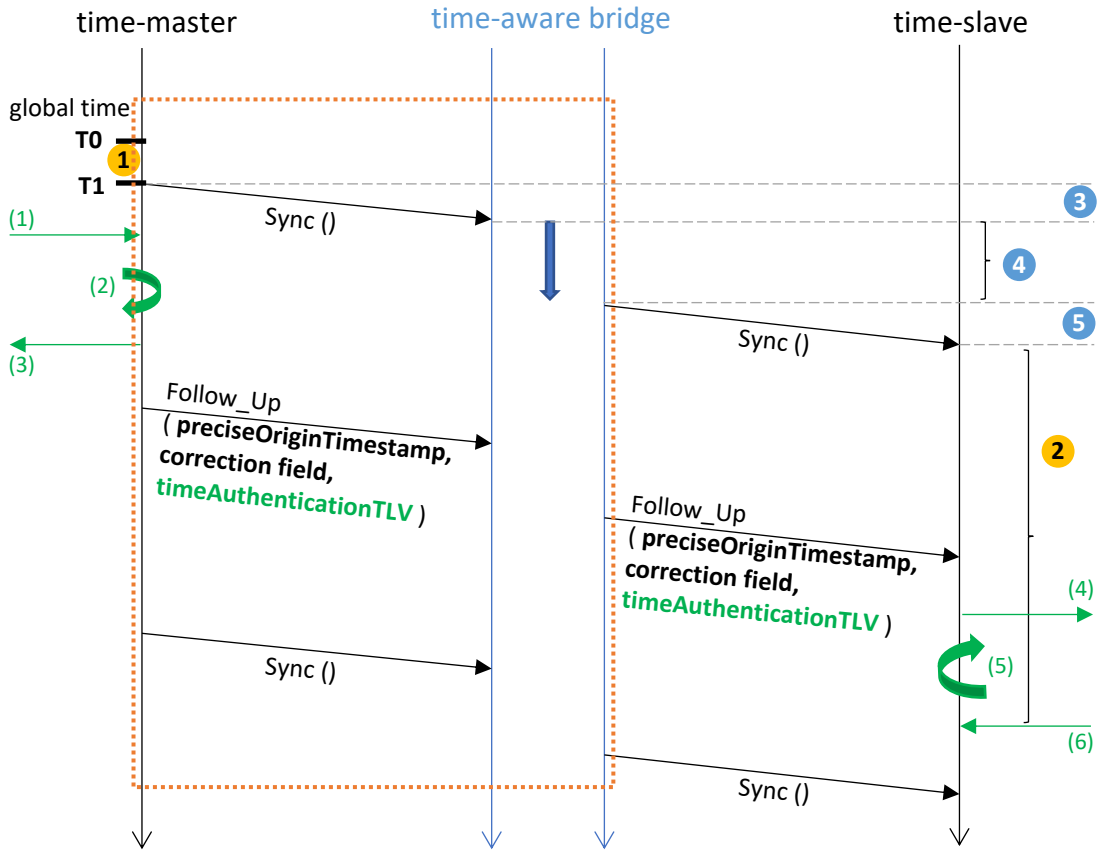
Integrated Security Mechanism

Ethernet – Secure Global Time Synchronization

- **AUTOSAR Sub-TLV : Time Authenticated** in Follow_Up message
  - Format

Frame Payload (max.1500 Bytes)											
PTP Header	Time Stamp	Follow_Up INFO TLV (IEEE)	Follow_Up INFO TLV (AR)	0 or more sub-TLVs	AR TLV: Sub-TLV Time Authenticated						
					T	L	Flags	Seq.Nr	FV.Len	FV	ICV

    - ICV secures the marked Follow_Up message
  - Process
    - (1) Assemble the Follow_Up message without ICV
    - (2) Compute the ICV
    - (3) Append the ICV to Follow_Up message
    - (4) Disassemble the Follow_Up message from ICV
    - (5) Verify the ICV of Follow_Up message
    - (6) Provision the global time and/or offset time to the customers
- ICV generation, ICV verification timeout for steps (2), (5) respectively
- Real/Fake Sync detection
  - Time slave to detect any violation to sequence [SYNC, Follow_UP] within Follow_Up timeout
  - Time slave to detect when Follow_Up message is received before the rx-debounce-time



global time at slave = T0 + 1 + delay + 2

delay = propagation delay (3 + 5) + forwarding delay 4

Integrated Security Mechanism

CAN – Secure Global Time Synchronization

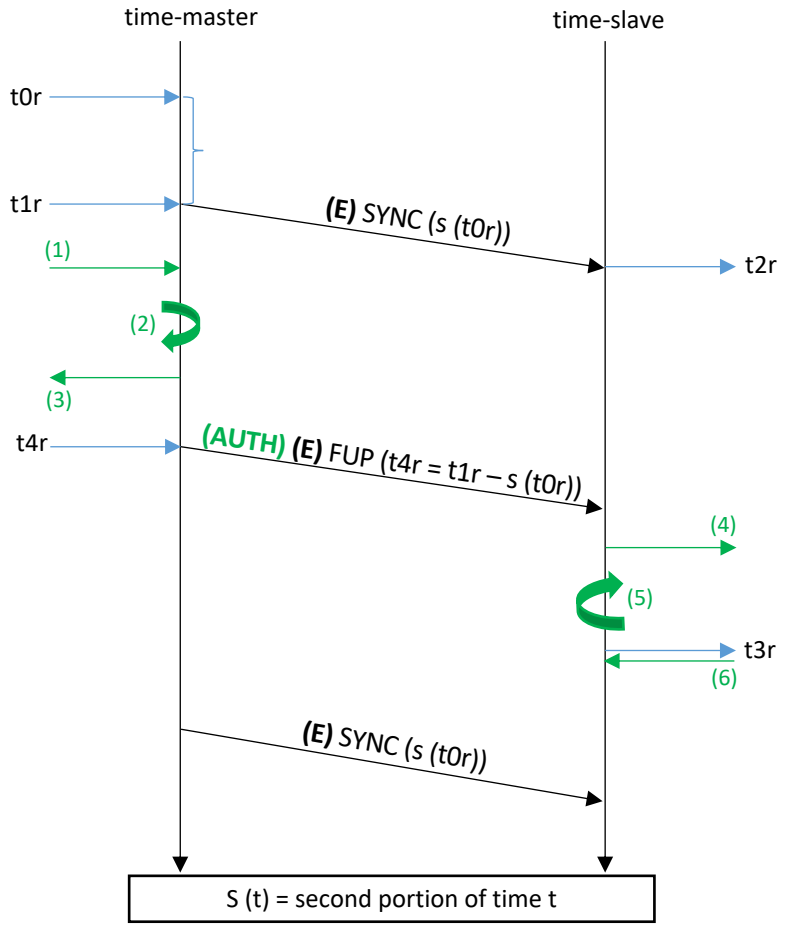
- **Authenticated Format** of extended Follow-Up (FUP) message
  - Format

Frame Payload (max.64 Bytes)											
B-0	B-1	B-2	B-3	B-4	B-5	B-6	B-7	B-8	B-9	B-10	B-11 to B-63
0x78	UB2	D, SC	Flags	SyncTimeNSec				FVL	ICVL	FV	ICV
0x88	CRC	D, SC	Flags	SyncTimeNSec				FVL	ICVL	FV	ICV

  - Process
    - (1) Assemble the FUP message without ICV
    - (2) Compute the ICV
    - (3) Append the ICV to FUP message
    - (4) Disassemble the FUP message from ICV
    - (5) Verify the ICV of FUP message
    - (6) Provision the global time to the customers
  - Same process is followed to secure the extended offset synchronization (OFS) message

Frame Payload (max.64 Bytes)											
B-0	B-1	B-2	B-3	B-4	B-5	B-6	B-7	B-8	B-9	B-10	B-11 to B-63
0x78	UB2	D, SC	Flags	SyncTimeNSec				FVL	ICVL	FV	ICV
0x88	CRC	D, SC	Flags	SyncTimeNSec				FVL	ICVL	FV	ICV

- ICV generation, ICV verification timeout for steps (2), (5) respectively
- Real/Fake Sync detection
  - Time slave to detect any violation to sequence [SYNC, FUP] within FUP timeout
  - Time slave to detect when FUP message is received before the rx-debounce-time



computed rel. time = (t3r-t2r) + s(t0r)+ t4r

Limitation:

SGTS is supported only on CAN FD channel.

- The integrated security mechanism is too complex to achieve on classic CAN busses due to payload limitation, therefore any incorporated solution will leave security vulnerabilities (e.g., cryptographic attacks, DoS).
- Today’s ECUs in the vehicle E/E architecture, support both classic CAN and CAN FD channels.

Integrated Security Mechanism

FlexRay – Secure Global Time Synchronization

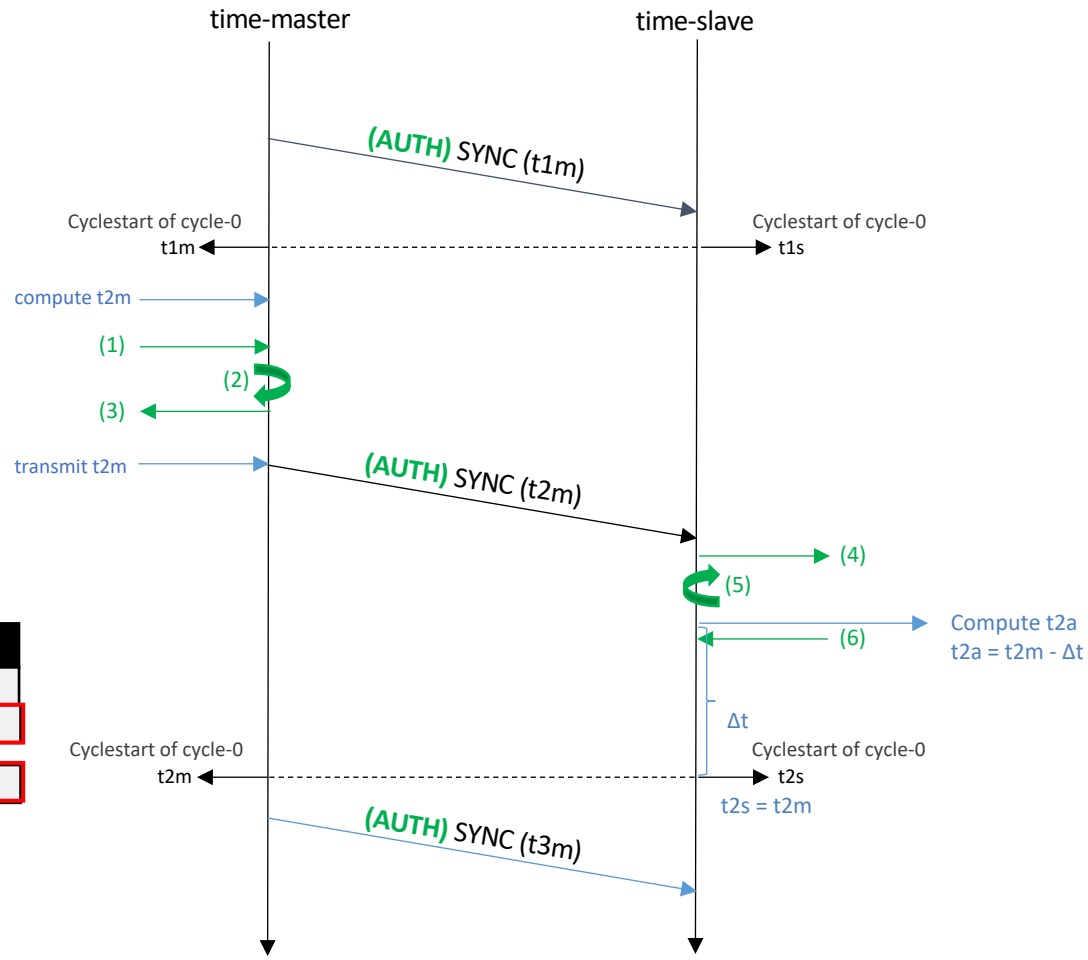
- **Authenticated Format** of time synchronization (SYNC) message
  - Format

Frame Payload (max.254 Bytes)										
B-0	B-1	B-2	B-3	B-4	B-5	B-6 to B-15	B-16	B-17	B-18	B-19 to B-253
0x50	UB2	D, SC	Flags	UB1	UB0	SyncTimeSec, SyncTimeNSec	FVL	ICVL	FV	ICV
0x60	CRC	D, SC	Flags	UB1	UB0	SyncTimeSec, SyncTimeNSec	FVL	ICVL	FV	ICV

  - Process to secure the SYNC message
    - (1) Assemble the SYNC message without ICV
    - (2) Compute the ICV over assembled SYNC message data
    - (3) Append the ICV to SYNC message
    - (4) Disassemble the SYNC message from ICV
    - (5) Verify the ICV of SYNC message
    - (6) Provision the global time to the customers
  - Same process is applied to secure the OFS message

Frame Payload (max.254 Bytes)											
B-0	B-1	B-2	B-3	B-4	B-5	B-6, B-7	B-8 to B-15	B-16	B-17	B-18	B-19 to B-253
0x34	UB2	D, SC	Flags	UB1	UB0	RS	OFSTimeSec, OFSTimeNSec	FVL	ICVL	FV	ICV
0x44	CRC	D, SC	Flags	UB1	UB0	RS	OFSTimeSec, OFSTimeNSec	FVL	ICVL	FV	ICV

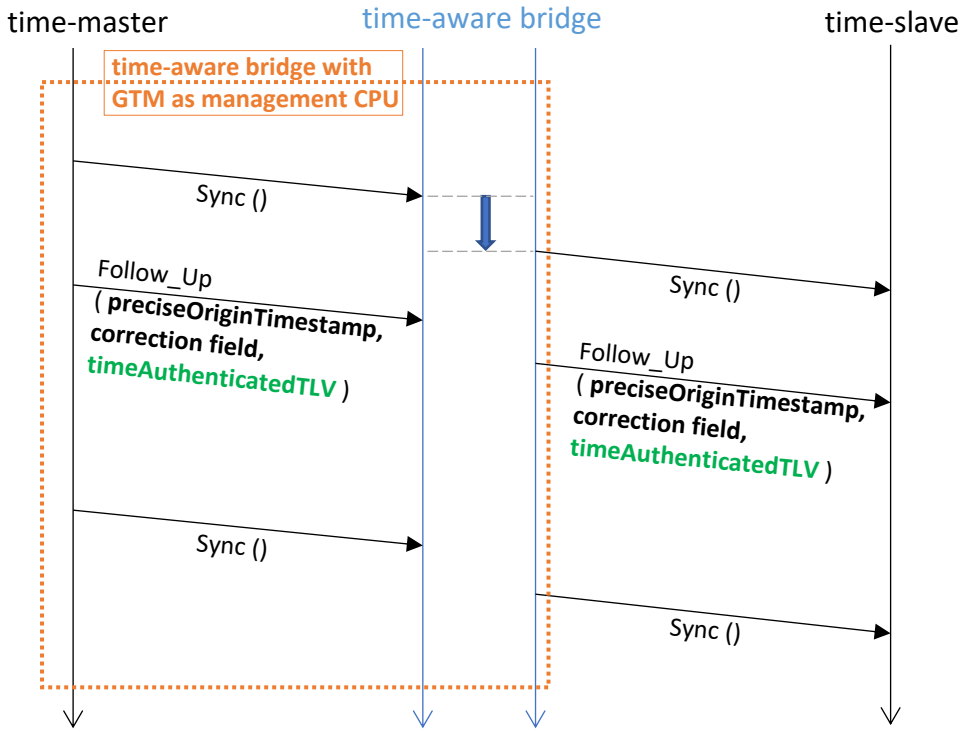
- ICV generation, ICV verification timeout for steps (2), (5) respectively

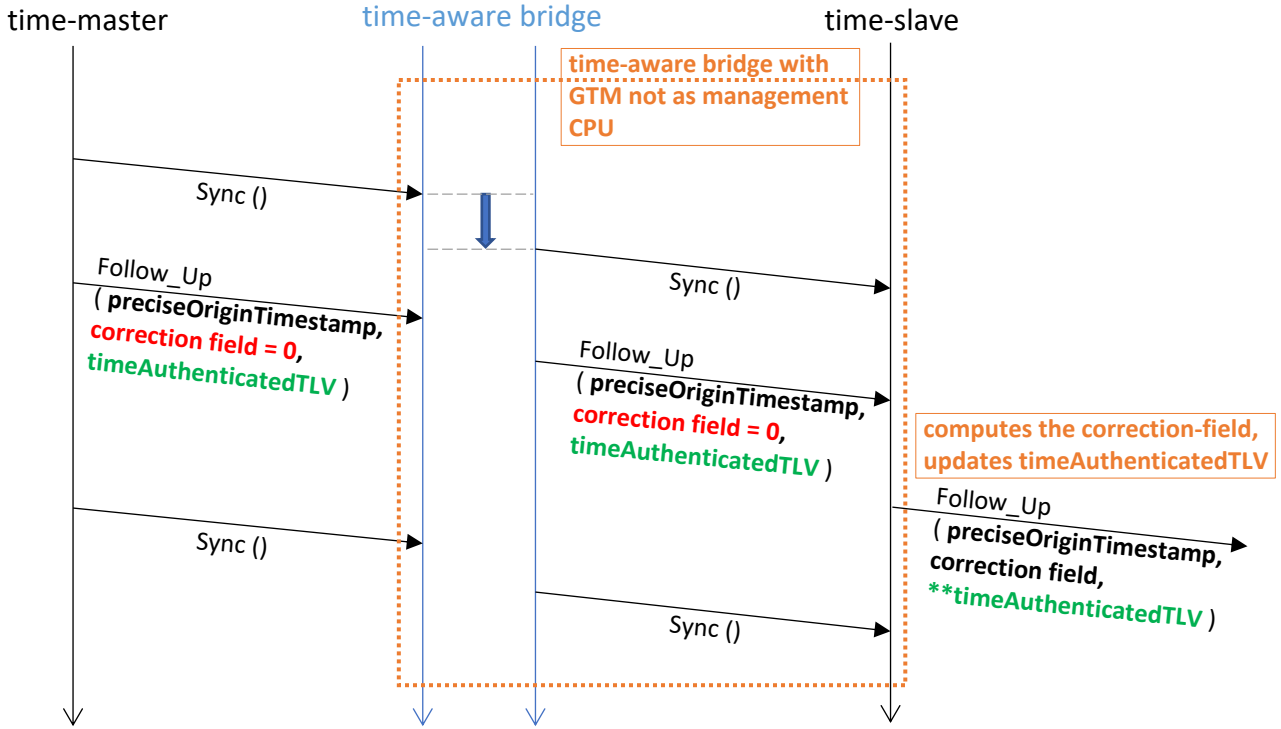


Integrated Security Mechanism - Ethernet

Additional Considerations

- Correction field (forwarding delay/residence time) protection

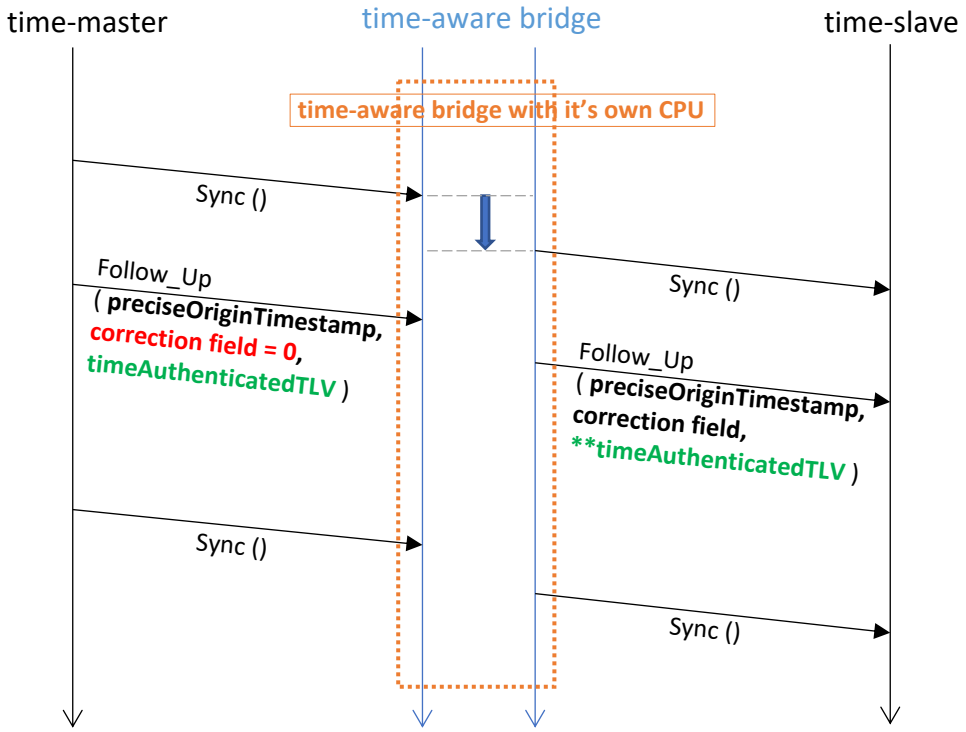




Integrated Security Mechanism - Ethernet

Additional Considerations

- Correction-field can be updated
  - in time-master, time-slave or time-aware bridge based on system design
  - several times based on number of time-aware bridges in system
- Solutions for the protection of correction-field
  - Solution-1 (verify and update, at every bridge)
    - Verify the timeAuthenticatedTLV
    - Update the correction-field
    - Update the timeAuthenticatedTLV
  - Solution-2 (verify and add, at every bridge)
    - Verify the timeAuthenticatedTLV
    - Update the correction-field
    - Add the new timeAuthenticatedTLV with updated correction-field
- By evaluating the attribute ‘complexity’ in each solution
  - Solution-1 is chosen



Integrated Security Mechanism - Ethernet

Additional Considerations

- Pdelay (path/propagation delay) protection
  - Automotive environment as ‘Closed System’
    - Propagation delay is static value set in the production or service phase and otherwise does not change.
    - Audio Video Bridging (AVB) use case
      - In order to maintain the timing accuracy in case of situations where the vehicle has undergone repair, replacement of parts or wiring changes, the static value of propagation delay needs to be updated. Dynamic calculation of propagation delay via Pdelay protocol is used in this case.
    - Non-AVB use case
      - Propagation delay is not needed to dynamically calculate.
  - Automotive environment with ‘Plug-and-Play devices’
    - Not covered as part of gPTP [802.1as-2011] standard

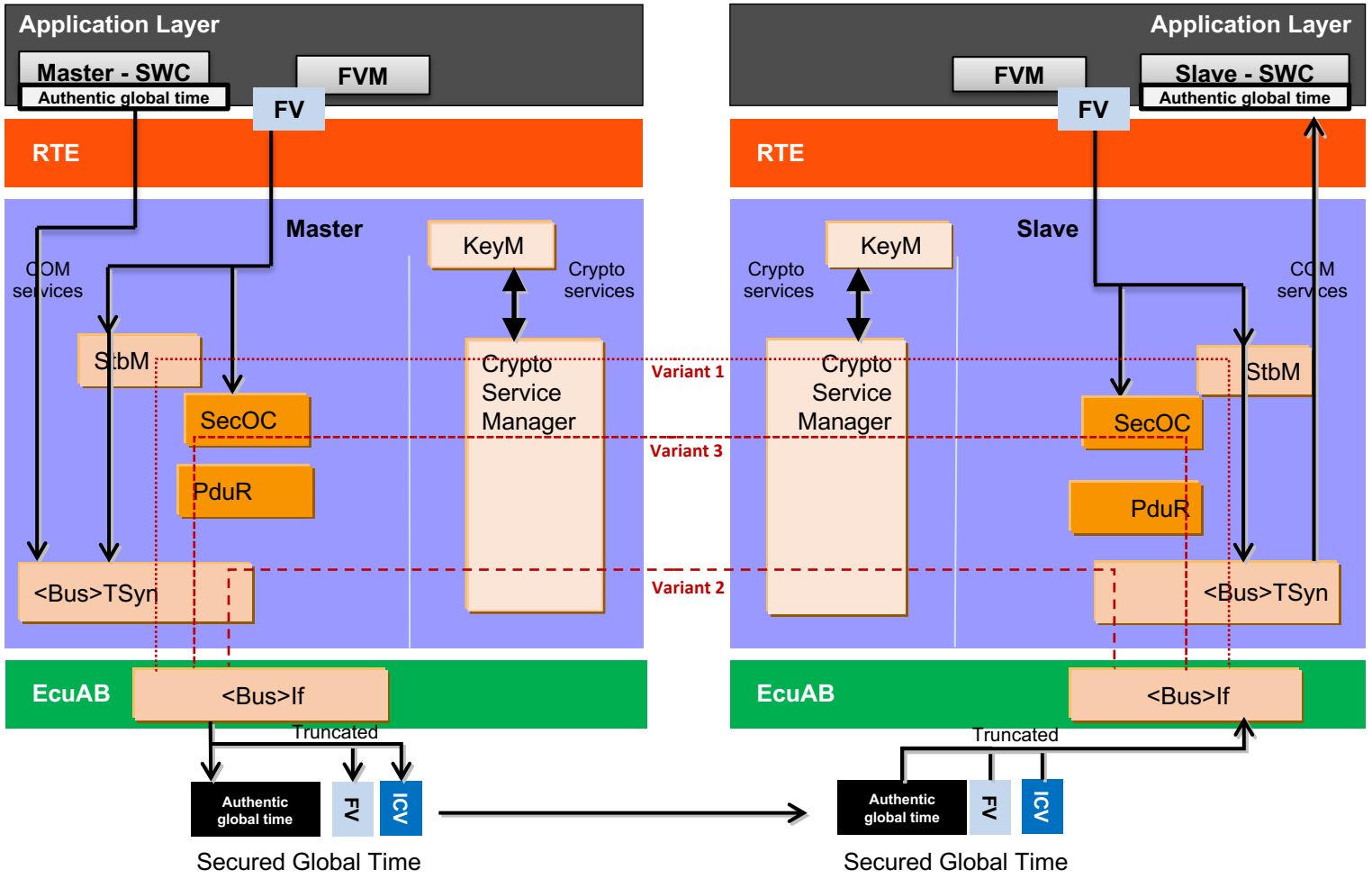
- Pdelay protocol messages are not protected via integrated security mechanism
  - Plausibility check shall ensure propagation delay is within the boundary values.

Architecture

Software Architecture Variants

- Variant of software architecture are derived based on module that interfaces the crypto stack. The respective module shall also interface the FVM to fetch FV.

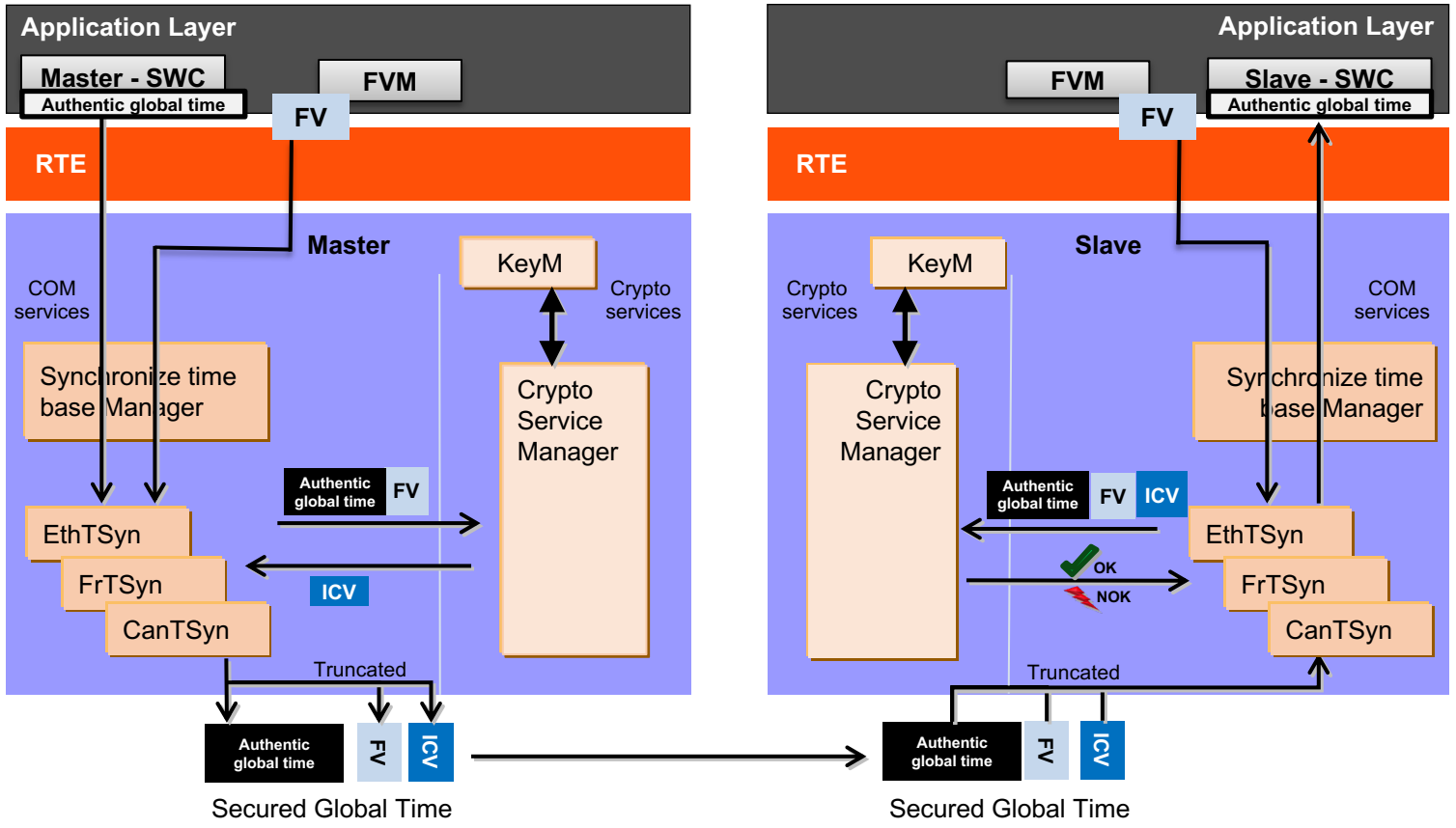
- Variant 1
  - StbM based architecture
- **Variant 2**
  - **<Bus>TSyn based architecture**
- Variant 3
  - SecOC based architecture



Architecture

Software Architecture – Variant2 – <Bus>TSyn based architecture

- <Bus>TSyn modules interface the crypto stack (CSM module)
  - ICV generation process
    - <Bus>TSyn construct the authentic global time messages
    - <Bus>TSyn coordinates the ICV generation
      - Fetches the FV from FVM via StbM
      - Invokes CSM to generate ICV
    - <Bus>TSyn construct secure global time messages and trigger the transmission
  - ICV verification process
    - <Bus>TSyn coordinates the ICV verification
      - Fetches the FV from FVM via StbM
      - Invokes CSM to verify ICV
- <Bus>TSyn modules handle the ICV generation/verification timeouts



Architecture

Software Architecture Variants – Evaluation

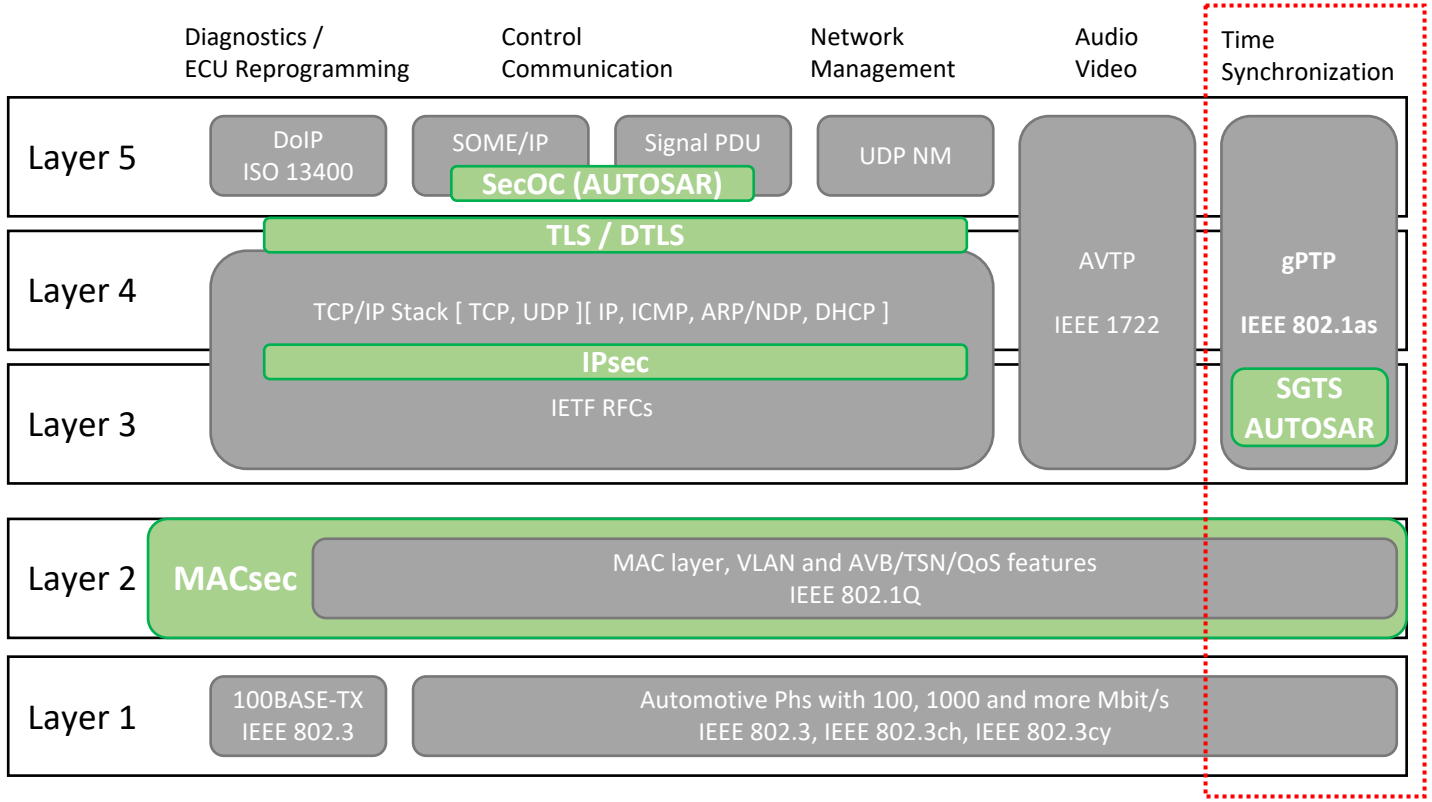
Attributes			Variant [StbM based]	Variant 2 [<Bus>TSyn based]	Variant 3 [SecOC based]
Methodology	New communication paths	• Non-secure path → GeneralPurposePdu/IPdu • Secure path → SecuredIPdu ↔ IPdu • Communication path → EthIf ↔ PduR	-	-	☹
Development / Integration	Static Implementation	Implementation overaheads (⇧)	☹	-	-
	Toolchain update	Number of modules need major changes (⇧)	☹	☹ ☹	☹ ☹ ☹
	Configuration	Number of modules need major changes (⇧)	☹	☹ ☹	☹ ☹ ☹
		Configuration consistency across number of modules across several usecases (⇧) (*1)	☹	-	☹ ☹
	Maintenance	Impact due to changes in crypto stack (Interfaces, behaviour)⇧	☹	☹ ☹	-
Operations	RunTime Impacts	• Synchronization Point Precision (⇩) • Potential vulerabilities in memory (⇧)	☹	-	☹ ☹
Backward Compatibility	Specification-wise		-	-	☹
	Bus-wise	Compatibility level 3	-	-	-
		Changes needed in ECUs without SGTS to stay compatible with ECUs with SGTS in a network (⇧)	☹	☹	☹ ☹
	Application-wise		-	-	-

(*1) consistency to CSM configuration across several use cases is applicable for all variants and not considered for this analysis

Other Security Mechanisms

External Security Mechanism - Ethernet

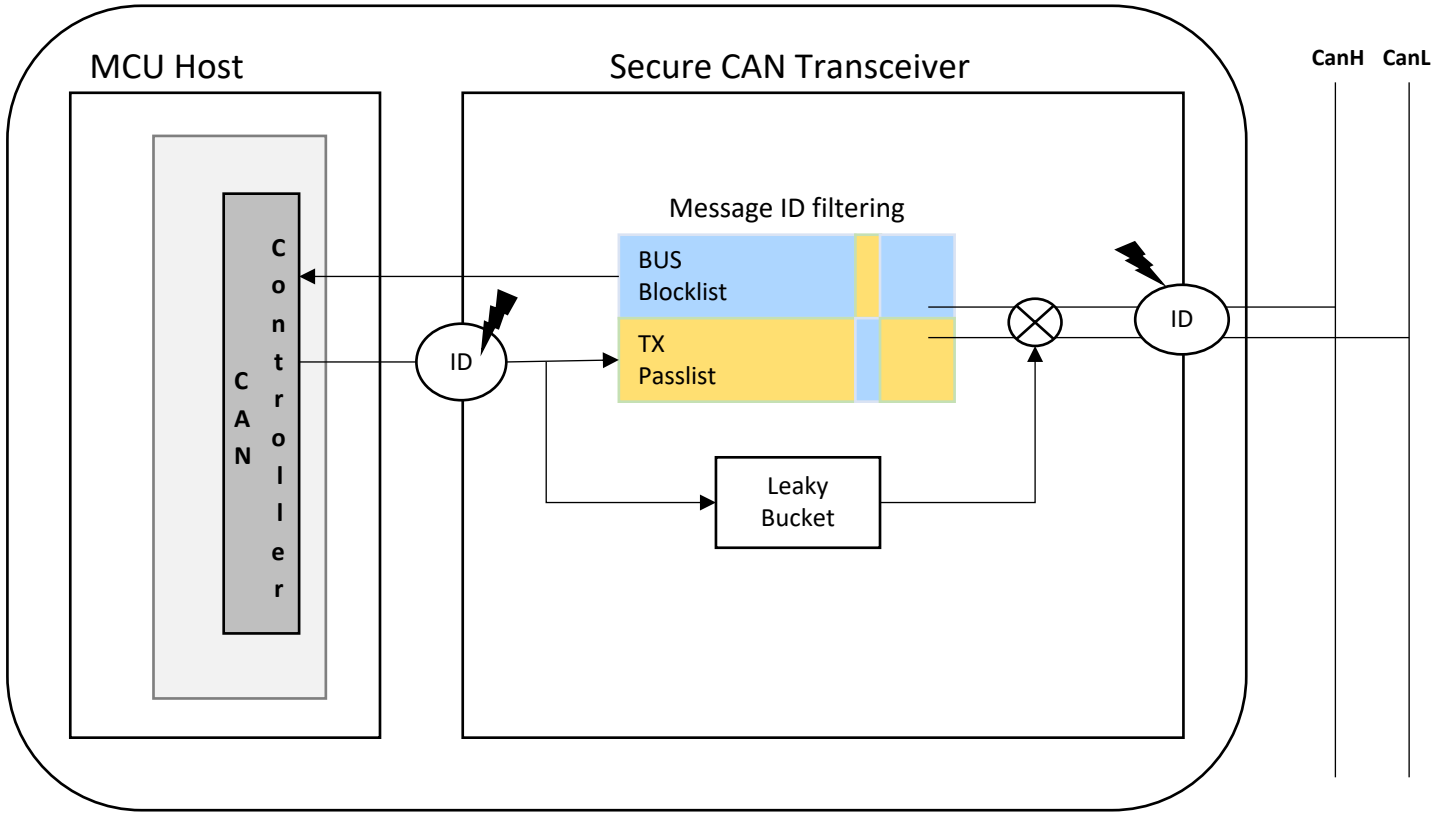
- MACsec (IEEE 802.1AE)
  - Provides the point-to-point security on Ethernet links
  - MACsec in automotive domain is in process of standardization
  - Research to find whether all automotive use cases and constraints can be satisfied (e.g., system startup time, faster time synchronization at startup, ..) is still ongoing for MACsec-capable switches
- IPsec and DTLS
  - Not a solution to secure global time. AUTOSAR supports PTP over IEEE 802.3 only.
  - IEEE 1588 supports PTP over UDP



Other Security Mechanisms

External Security Mechanism - CAN

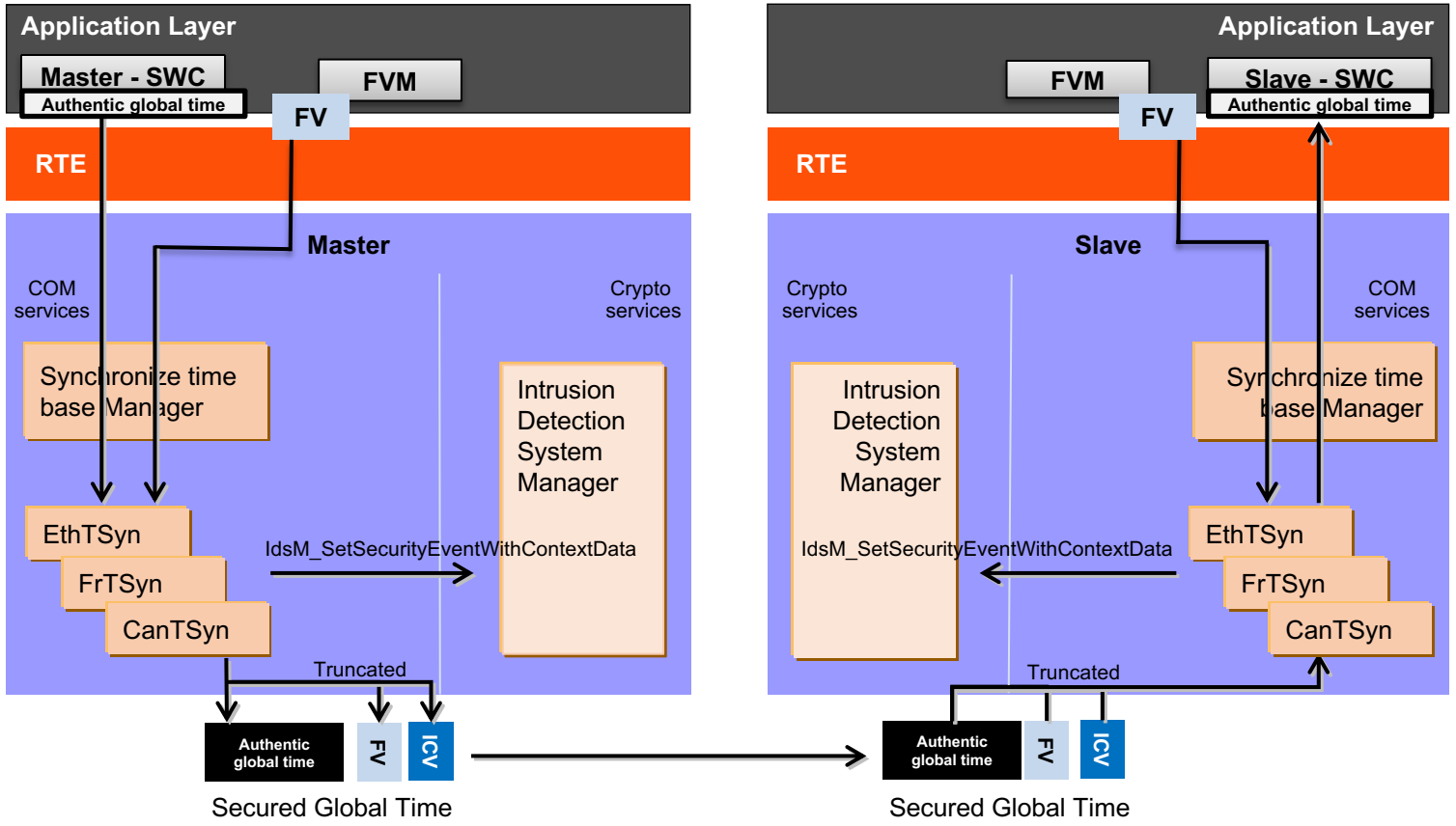
- Secure CAN Transceiver
  - Performs detection and containment of security incidents → flooding, tampering and spoofing at physical layer
- Fingerprints the transmitting ECU (via clock skew, voltage values …) to authenticate the source
  - Physical characteristics tend to change due to environment factors like temperature, aging of components; therefore, fingerprinting may fail.
  - Fails to detect the malicious messages from the software layers of the compromised ECU, as there will be no changes to the signal characteristics.



Other Security Mechanisms

Monitoring and Management

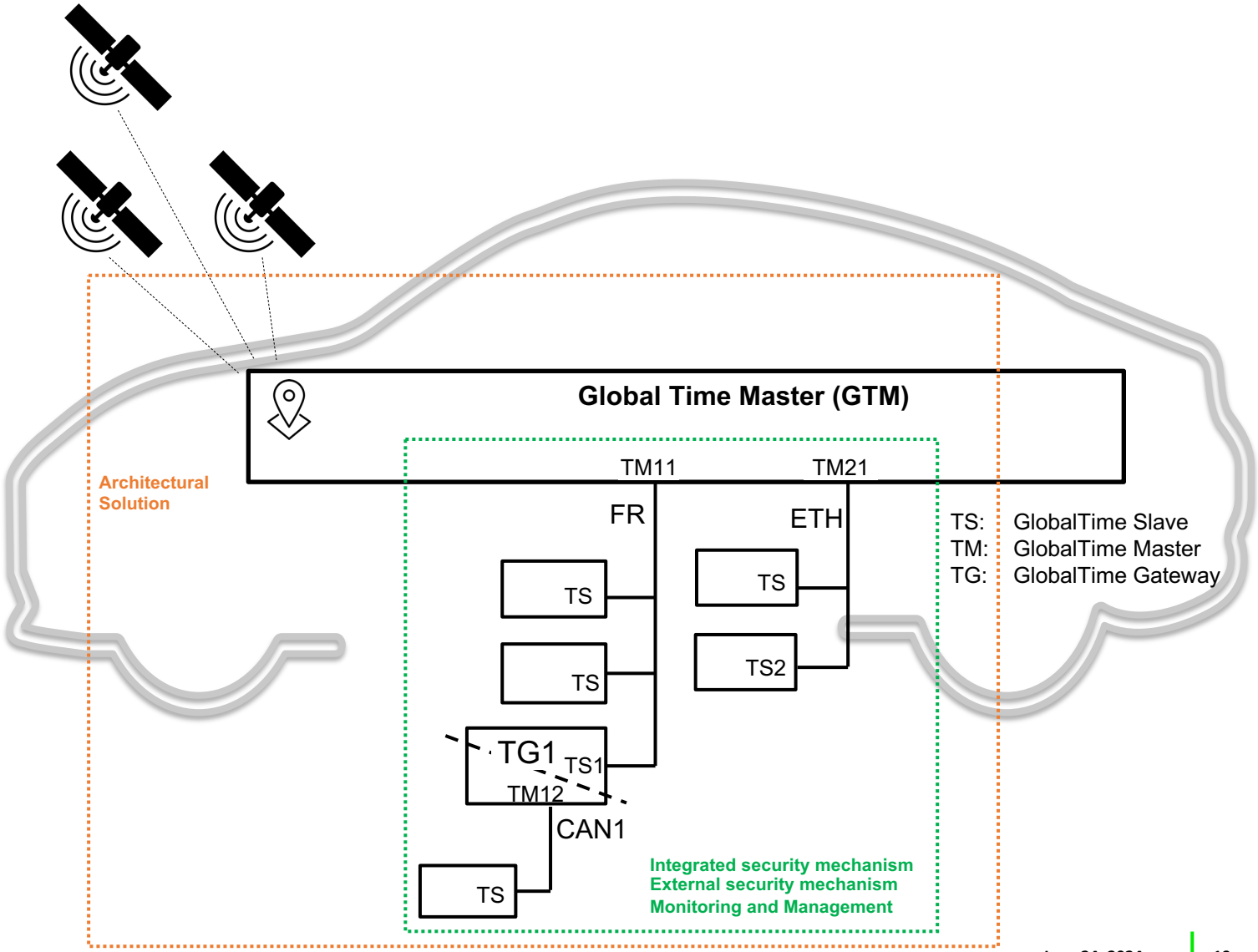
- Intrusion Detection
  - Any suspicious activities observed during global time synchronization shall be detected and reported to IDS
  - Below security events are detected in <Bus>TSyn modules of time-master
    - *<Bus>TSYN_SEV_FRESHNESS_NOT_AVAILABLE*
    - *<Bus>TSYN_SEV_ICV_GENERATION_FAILED*
  - Below security events are detected in <Bus>TSyn modules of time-slave
    - *<Bus>TSYN_SEV_FRESHNESS_NOT_AVAILABLE*
    - *<Bus>TSYN_SEV_ICV_VERIFICATION_FAILED*
    - *<Bus>TSYN_SEV_SYNC_FOLLOWUP_SEQUENCE_ERROR*



Other Security Mechanisms

Architectural Solution

- Securing the source of time to vehicle via
  - Protection mechanism
  - Intrusion Detection mechanism
  - Redundant time sources
- Security qualifier via User Data
  - Notify the time-slave when the global time is not managed from secure source
- Plausibility checks
  - Time Slaves to ensure the global time from time-master is within boundaries
- Safety
  - No security → No safety



Challenges to Solve

Holistic solution to Secure Global Time

Effective Configuration

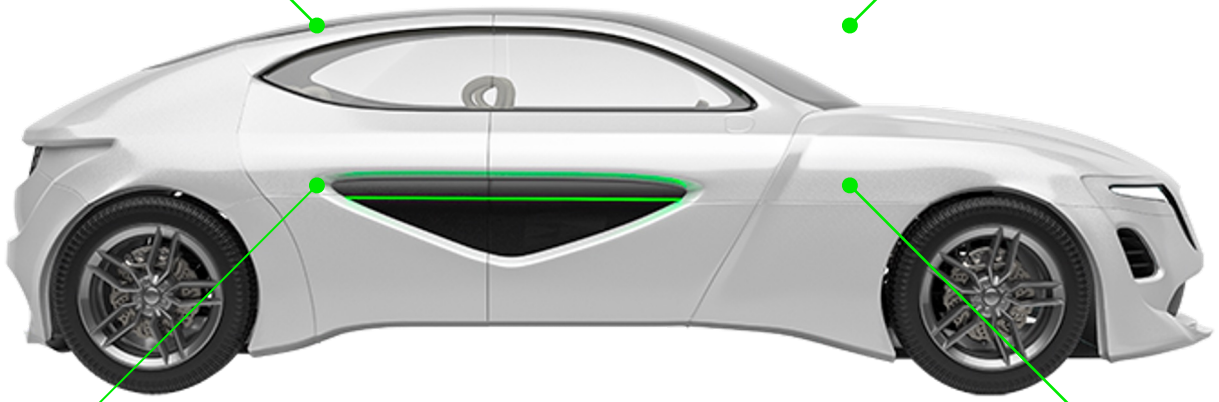
….

Zero impact on Precision of global time
No room for security vulnerabilities

Integrated v/s External

….

MACsec, Secure CAN transceiver replace the integrated security mechanism?



Holistic Solution

…

Extension to Adaptive platform
Threat analysis & Risk Assessment to reach security with best cost

Futuristic

….

Time Synchronization between backend and vehicle

Contact us

tarav.shah@elektrobit.com
Pavithra.Kumaraswamy@elektrobit.com
https://elektrobit.com