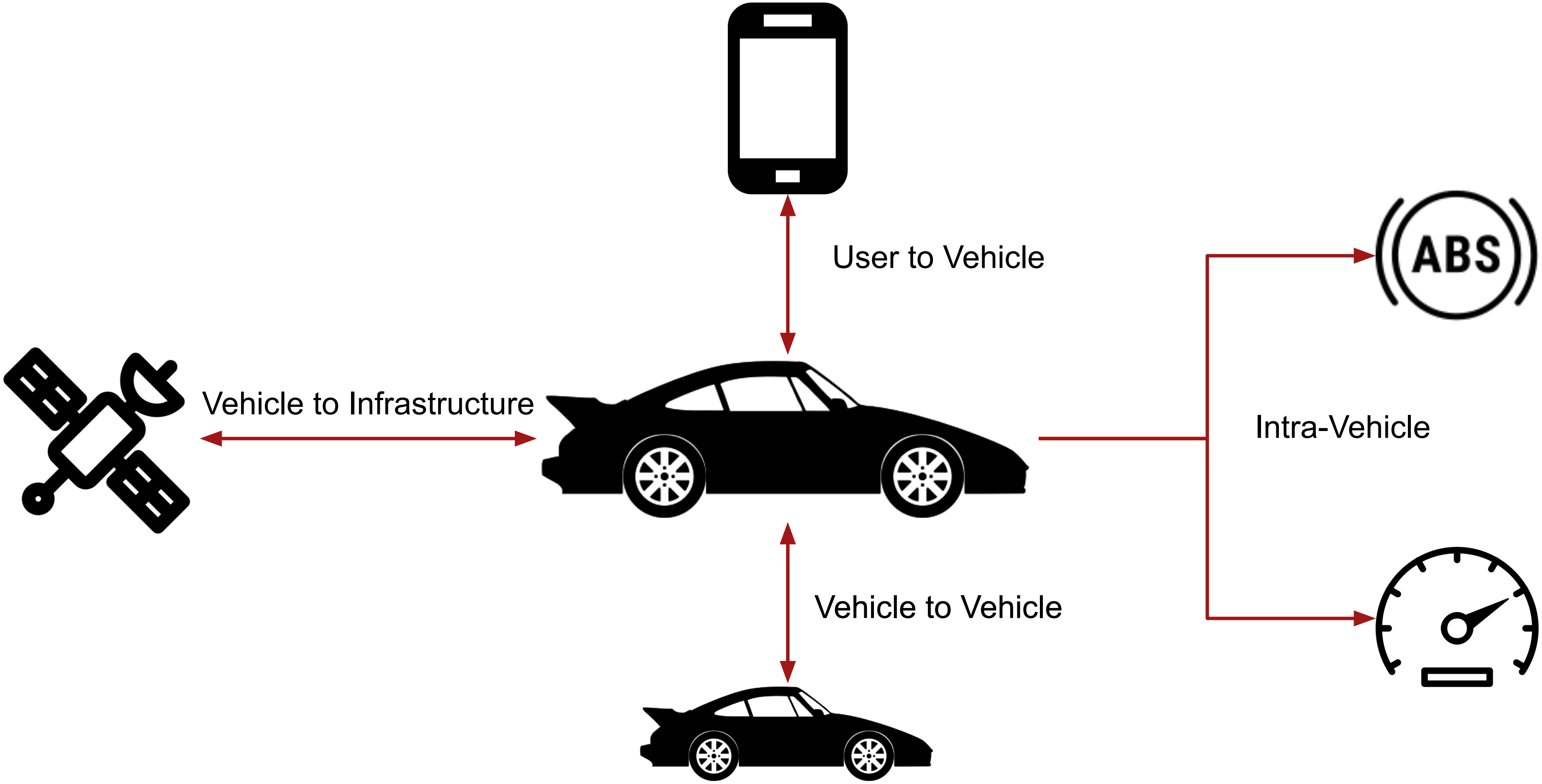
EDCC 2020

CINNAMON: A Module for AUTOSAR Secure on-board Communication

Giampaolo Bella ✻, Pietro Biondi ✻, Gianpiero Costantino✦, Ilaria Matteucci✦
giamp@dmi.unict.it , pietro.biondi@phd.unict.it , gianpiero.costantino@iit.cnr.it , ilaria.matteucci@iit.cnr.it
✻ Dipartimento di Matematica e Informatica, Università di Catania, Italy
✦ Istituto di Informatica e Telematica,Consiglio Nazionale delle Ricerche, Pisa, Italy

Automotive communication domains



Introduction

Controller area network (CAN-bus):

- Intra-Vehicular communication standards
- Serial communication protocol
- Message anti-collision protection
- Error detection

Weaknesses: confidentiality, authentication, integrity

A **DBC** stores the mapping between CAN frame payloads and functionalities of a vehicle, as decided by the Original Equipment Manufacturer.

Threat Model

The attacker in general aims to mount the following attacks:

- **Replay:** re-use of valid CAN frames with malicious or fraudulent aims
- **Tampering:** manipulation of CAN frames to spoil their contents so that a receiving ECU cannot perform the operation that was originally meant.
- **Forging:** generation of a valid CAN frame, which is then able to generate a valid signal and activate a specific ECU functionality.
- **Fuzzing:** injection of CAN frames, which were previously forged, with the aim of studying the behaviour of a target ECU against unexpected inputs.
- **Masquerading:** misinterpretation of attacker’s identity by using a CAN ID of some other genuine ECU, thereby masquerading as that ECU.
- **Information Gathering:** identification of critical contents from CAN frames, such as the frame ID or payload and its associated ECU functionality, with the aim of using it against a target ECU to perform a post-attack.

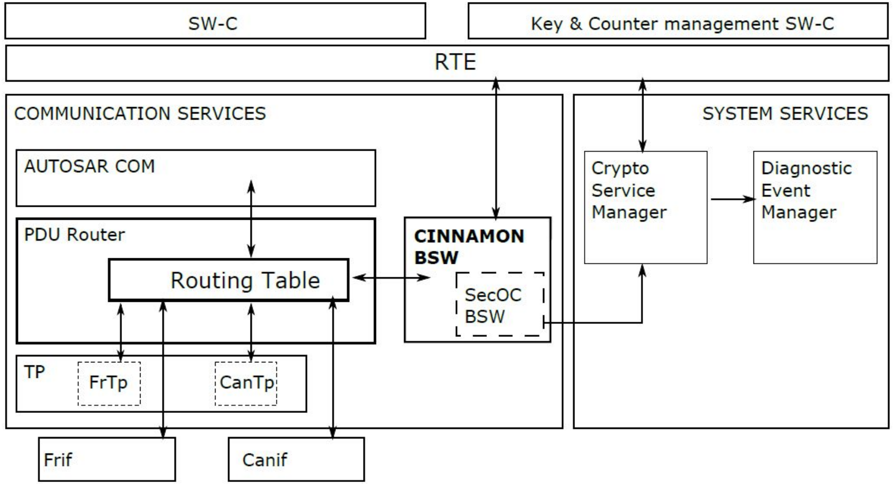
This paper introduces the **CINNAMON** module, whose requirements leverage and extend those already provided by SecOC(AUTOSAR).

CINNAMON insists not only on authenticity and integrity (as SecOC does) but also on confidentiality of CAN bus communications.

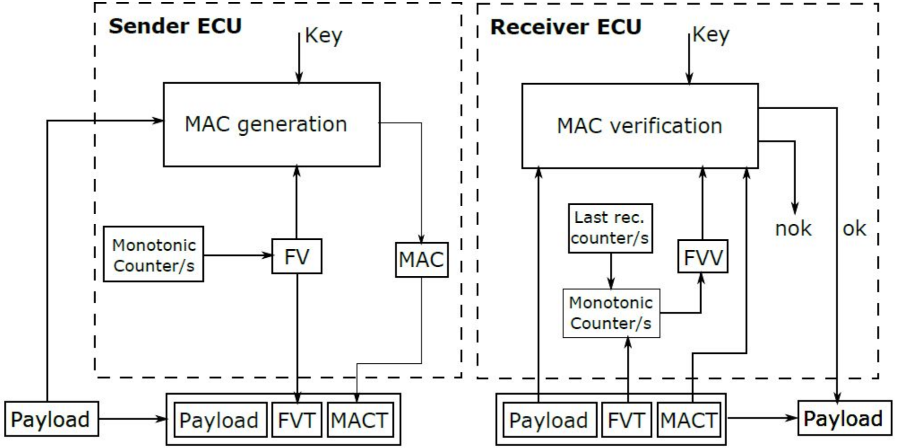
CINNAMON Requirements

1. Functional → Configuration of different security properties
2. Initialisation → Initialisation of security information
3. Normal Operations → Support of Automotive BUS Systems
4. Normal Operations → Support of capability to extract Authentic frame without Authentication
5. Non-Functional → Authentication and verification processing time
6. Support for end-to-end and point-to-point protection

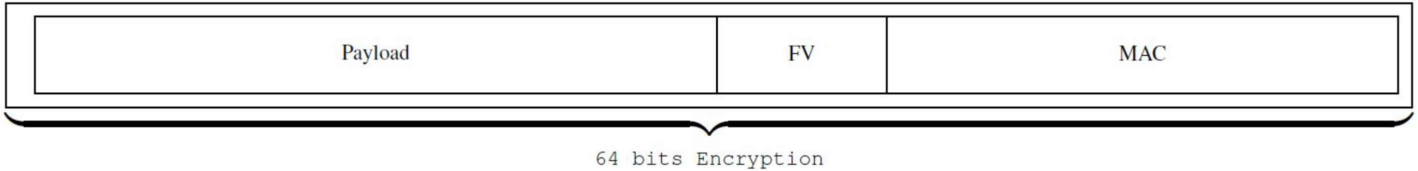
CINNAMON Specification



Integrating CINNAMON BSW module in AUTOSAR



MAC Generation and Verification

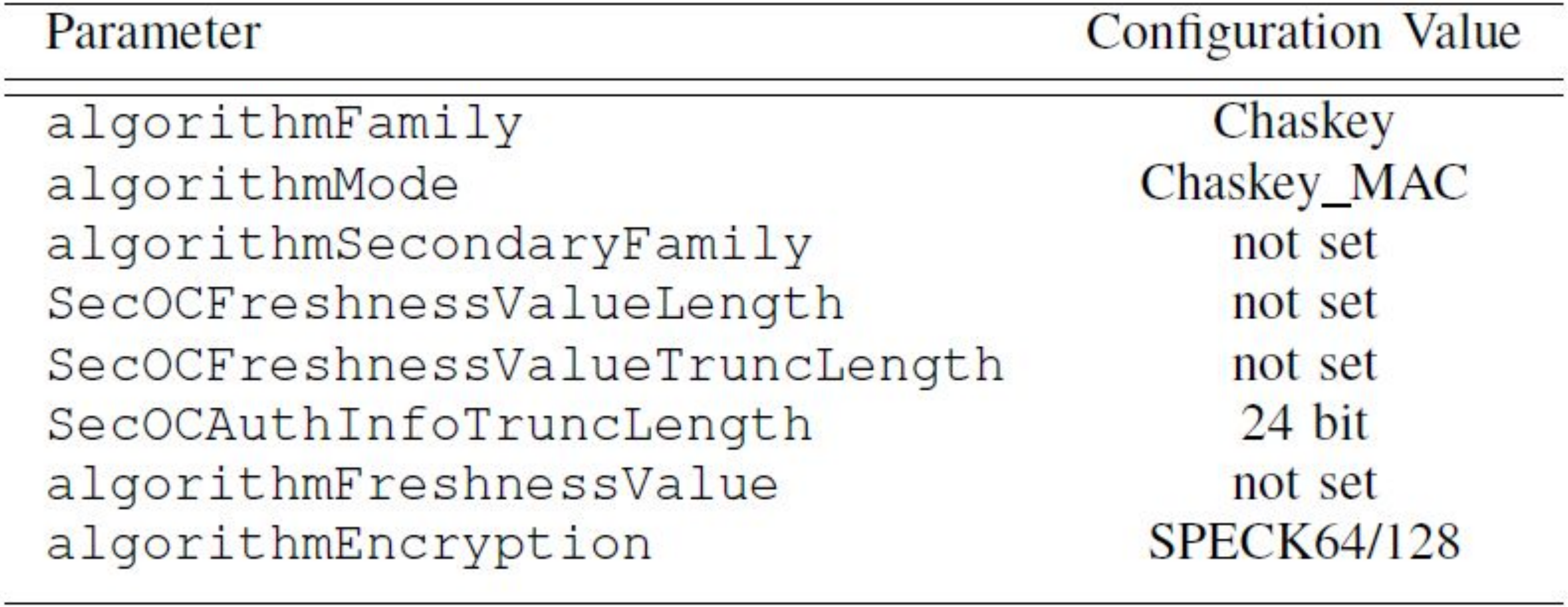


The CINNAMON Secured CAN Data field

CINNAMON Security Profiles

- **algorithmFamily:String [0..1]** This parameter identifies the family of authentication algorithms.
- **algorithmMode:String [0..1]** This parameter identifies which MAC algorithm of the family is used.
- **algorithmSecondaryFamily:String [0..1]** This parameter identifies a secondary family of authentication algorithms, if any.
- **authInfoTxLength:PositiveInteger** denotes the length of the truncated MAC.
- **freshnessValueLength:PositiveInteger** denotes the length of the generated freshness value.
- **freshnessValueTruncLength:PositiveInteger** denotes the length of the truncated freshness value inserted in a frame.
- **algorithmFreshnessValue:String [0..1]** denotes the algorithm used to generate the freshness value.
- **algorithmEncryption:String [0..1]** denotes the encryption algorithm.

CINNAMON Security Profiles (cont.)



Parameter	Configuration Value
algorithmFamily	Chaskey
algorithmMode	Chaskey_MAC
algorithmSecondaryFamily	not set
SecOCFreshnessValueLength	not set
SecOCFreshnessValueTruncLength	not set
SecOCAuthInfoTruncLength	24 bit
algorithmFreshnessValue	not set
algorithmEncryption	SPECK64/128

Example CINNAMON Security Profile

Security Assessment

TABLE I: Security Properties

Security Property	CINNAMON
Confidentiality	✓
Authentication	✓
Integrity	✓
Freshness	✓

TABLE II: Mitigated Threats

Threats	CINNAMON
Replay	✓
Tampering	✓
Forging	✓
Fuzzing	✓
Masquerading	✓
Information Gathering	✓

Outline Of Prototype Implementation

Testbed:

- 2 STM32F407 Discovery boards, each with an ARM Cortex M4 processor
- USB-to-CAN interface

Implementation:

- **SPECK-64**: Symmetric cipher used in systems with low computational resources.
  - Block cipher with 64-bit block size
  - Supported key lengths: 128, 192 and 256 bit
  - Efficiency in software and hardware

- **Chaskey:** permutation-based MAC algorithm based on Addition-Rotation-XOR (ARX).
  - Efficient MAC algorithm for microcontrollers
  - It is intended for applications that require 128-bit security
  - Robustness under tag truncation

Performances:

Inexpensive hardware with 168 MHz clock.
Average of less than 6μs to generate or interpret a protected frame.

Conclusions and Future Work

- CINNAMON is an AUTOSAR compliant basic software module for confidentiality, integrity and authenticity on CAN bus.
- Compared to SecOC, CINNAMON avoids information gathering attacks
- CINNAMON is scalable in the sense that it can host additional security profiles that become necessary depending on the application domain
- Prototype implementation whose performances are promising on inexpensive hardware
- New security profiles and their implementation
- Extend CINNAMON to secure not only the CAN bus but also other buses

CINNAMON: A Module for AUTOSAR Secure on-board Communication

Thank you for your attention

Giampaolo Bella, Pietro Biondi, Gianpiero Costantino, Ilaria Matteucci
giamp@dmi.unict.it , pietro.biondi@phd.unict.it , gianpiero.costantino@iit.cnr.it , ilaria.matteucci@iit.cnr.it

https://sowhat.iit.cnr.it/