carroot: A Secure Automotive ECU for Connected Vehicles

Ahmer Raza

School of Computing

Overview

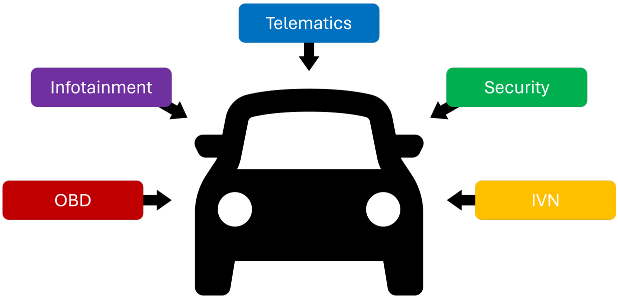
Introduction

Cars are becoming increasingly complex systems

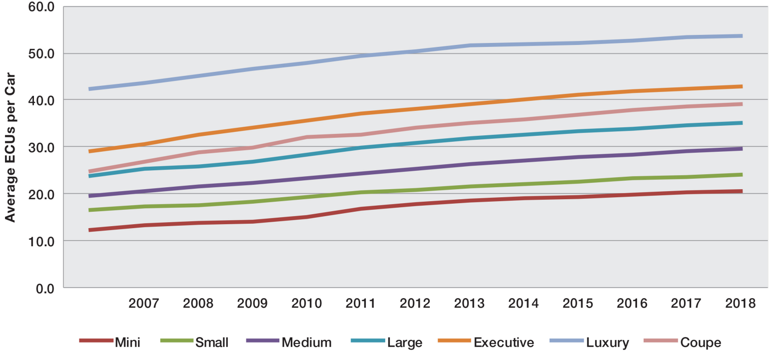
- Dozens of **Electronic Control Units** (ECUs) per car
- ECUs interconnected via **In-Vehicle Network** (IVN)

ECU security has not been a priority

- **Confidentiality, Integrity, Authenticity** (CIA) not provided



Introduction



Source: Strategy Analytics

Courtesy of Electronic Specifier

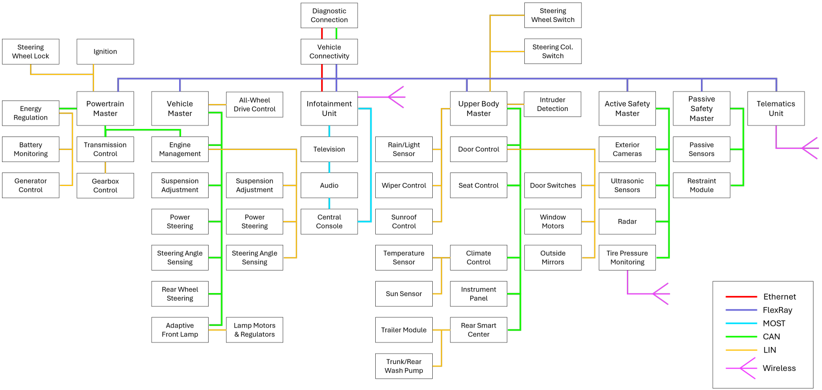
Introduction

We propose **carroot**

- **Trusted Execution Environment** (TEE) centric ECU design
- RISC-V-based Linux system

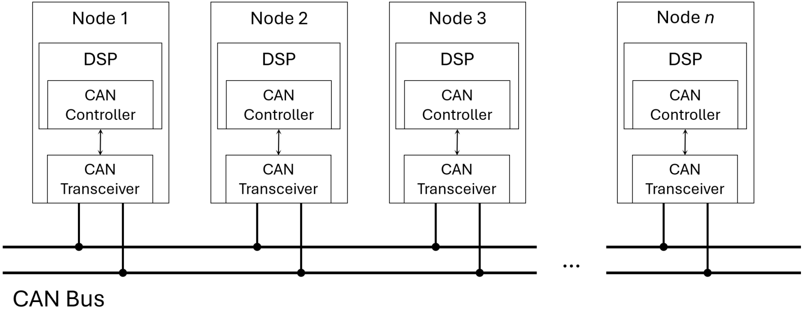
Background: Automotive Control

IVN is a **multiplexed medium**



Background: Automotive Control

Complex Area Network (CAN) is most widely used in IVNs



SOF	Identifier (11 bits)	RTR	IDE	r0	DLC	Data (64 bits)	CRC	ACK	EOF	IFS

Background: Automotive Control

IVNs are extremely vulnerable to attack

- Communication bus and wireless transmission are exploitable
  - Replay attacks and code injection
- Control over a single ECU means control over the whole IVN

Vehicles are critical to infrastructure

- Any vulnerability has potentially severe consequences

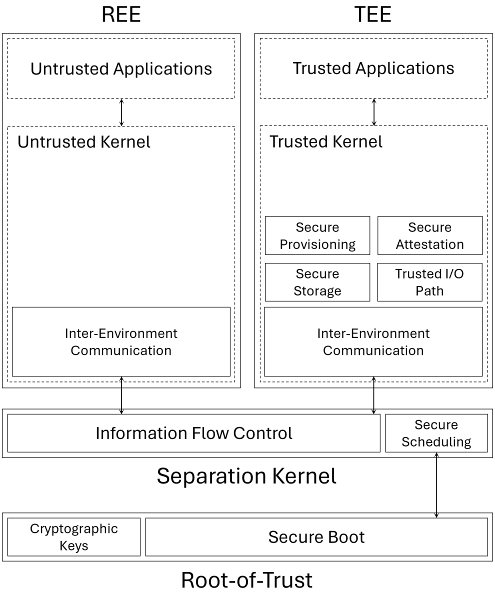
Background: Trusted Computing

Rich Execution Environment (REE): untrusted OS runs untrusted code on untrusted hardware

Trusted Execution Environment (TEE): provides isolated processing environment and security features

- Guarantees CIA of code, data, and runtime states
- **Secure boot** only allows bootstrapping trusted code
- **Root-of-Trust** (RoT) provides accurate **trust score**

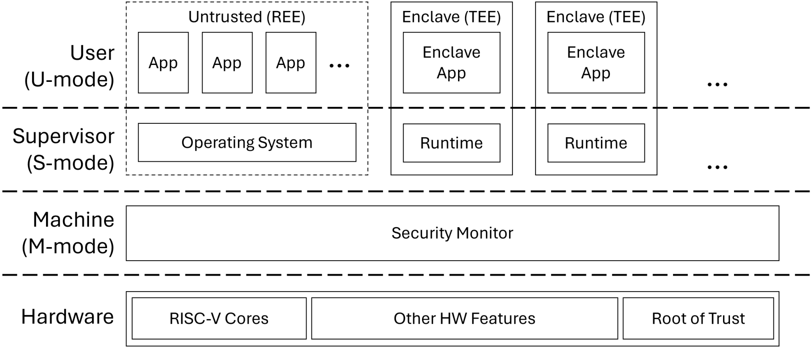
Background: Trusted Computing



Background: Trusted Computing

Keystone: open-source platform for architecting TEEs.
Consists of:

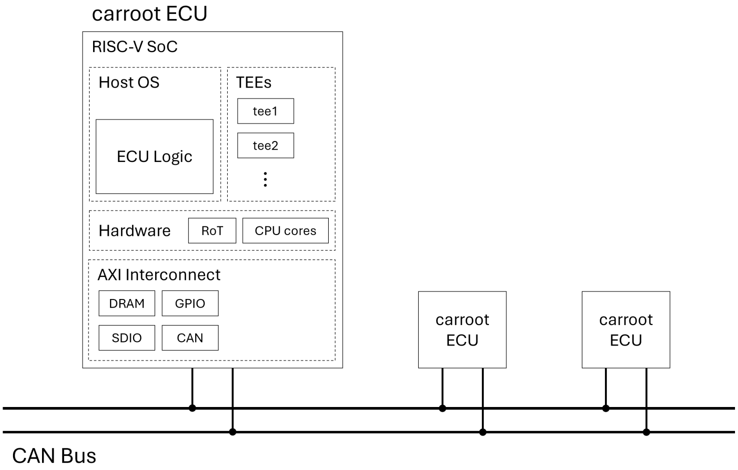
- Machine-mode **Security Monitor** (SM)
- Supervisor-mode **application runtime** (RT)
- User-mode **application development library** (SDK)



carroot: Proposed Platform

carroot: RISC-V System-on-Chip (SoC) ECU that uses TEEs

- Provides CIA for ECUs



carroot: Proposed Platform

carroot prevents threats in our threat model. For example:

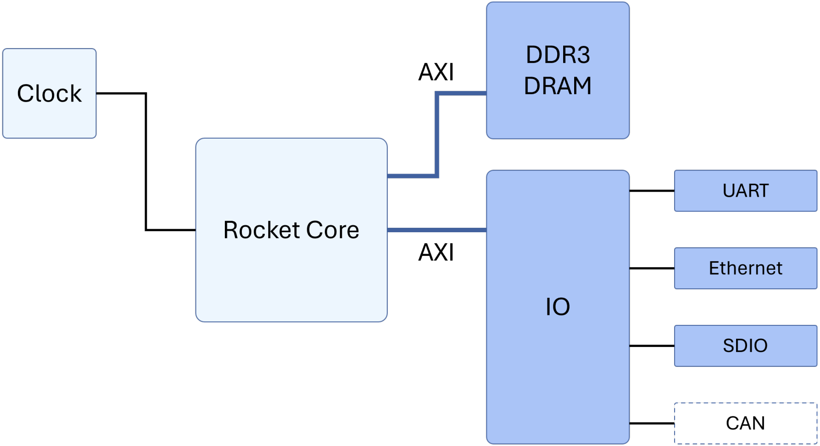
- Code injection (execution of untrusted code) is prevented by the TEE itself
- Replay attacks (strategic withholding and sending of messages) is prevented by guaranteed memory freshness

carroot: Physical Implementation

Partial prototype of carroot

- SoC consists of Rocket RISC-V softcore on Arty A7-100T FPGA board
  - AXI connects DRAM, UART, SDIO, etc.
- Rocket boot ROM modified to become a RoT
  - Secure boot procedure: cryptographic keys and SM measurement
- Keystone SM and related firmware added
  - Integrated into **Supervisor Binary Interface** (SBI)
- Debian Linux booted
  - Ongoing work to integrate Keystone RT and SDK

carroot: Physical Implementation



Conclusion: Future Directions

- Keystone SDK and RT incorporation
- Integration of CAN bus and related drivers
- Thorough individual system tests
- Testing of IVN with multiple carroot ECUs

References

[1] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, “Trends in Automotive Communication Systems,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, Jun. 2005, conference Name: Proceedings of the IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1435746

[2] S. Corrigan, “Introduction to the Controller Area Network (CAN),” Dallas, TX, Tech. Rep. SLOA101B, May 2016. [Online]. Available: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf

[3] M. Sabt, M. Achemlal, and A. Bouabdallah, “Trusted Execution Environment: What It is, and What It is Not,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug. 2015, pp. 57–64. [Online]. Available: https://ieeexplore.ieee.org/document/7345265

[4] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, “Keystone: an open framework for architecting trusted execution environments,” in *Proceedings of the Fifteenth European Conference on Computer Systems*, ser. EuroSys ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–16. [Online]. Available: https://dl.acm.org/doi/10.1145/3342195.3387532

[5] D. Lee, “Building Trusted Execution Environments,” Ph.D., University of California, Berkeley, United States – California, 2022, iSBN: 9798351476582. [Online]. Available: https://www.proquest.com/docview/2730786496/abstract/2DFD577A9C8F4005PQ/1

[6] C. Valasek and C. Miller, “A Survey of Remote Automotive Attack Surfaces,” Seattle, WA, Technical White Paper, Jul. 2014. [Online]. Available: https://ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf

[7] ——, “Adventures in Automotive Networks and Control Units,” Seattle, WA, Tech. Rep., 2014. [Online]. Available: https://ioactive.com/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf

[8] eugene tarassov, “eugene-tarassov/vivado-risc-v,” Apr. 2024, original-date: 2020-02-15T01:12:54Z. [Online]. Available: https://github.com/eugene-tarassov/vivado-risc-v

[9] “keystone-enclave/keystone,” May 2024, original-date: 2018-06-12T20:23:19Z. [Online]. Available: https://github.com/keystone-enclave/keystone