講演番号	157

Trusted Execution Environments in Automotive Real-Time Systems

Martin Manthe [1)] Dr. Philipp Jungklass [1)] Carolina Pelka [1)] Tim Kaiser [1)] Dr. Claude-Pascal Stober-Schmidt [1)] Dr. Dennis Kengo Oka Marco [2)] Marco Siebert [1)] Shingo Ise [2)]

1) IAV GmbH Ingenieurgesellschaft Auto und Verkehr, Carnotstraße 1, 10587 Berlin, Germany
2) IAV Japan Co., Ltd., Uchikanda Chuo Bldg. 3F, 1-18-13 Uchikanda, Chiyoda-ku, Tokyo, 101-0047, Japan

KEY WORDS: **Trusted Execution Environment, Hardware Security Module, Trusted Platform Module, ARM TrustZone, Software Defined Vehicle, Crypto Agility, Firmware Security Module**

„Electronic control units and software of increasing complexity have become essential to the operation of road vehicles in recent years. This software is often updated to increase functionality and maintain the safety and cybersecurity of road vehicles “. The fact that an ISO standard addresses this in its introductory chapter underlines the significance of the topic. New legal regulations on data protection (e.g., GDPR) or software updates (e.g., ISO 24089:2023-02), to functional requirements regarding online connectivity or customer experience, as well as technological challenges such as the imminent emergence of quantum computers in the foreseeable future and other requirements lead to specific technical tasks, such as securing confidential data, encrypting communication channels, or ensuring the update capability of security-relevant program components. Trusted Execution Environments can offer solutions for some of these problems. By isolating highly sensitive data and software components, they provide protection against manipulation and unauthorized access and thus make a substantial contribution to achieving the security objectives of confidentiality and integrity.

However, it is not only these supposedly new challenges that a control unit in a vehicle must meet. There are additional areas of requirements that strongly influence the selection of hardware components, particularly microcontrollers. Safety is of particular importance here, as it makes real-time capability of many critical vehicle functions absolutely necessary. Furthermore, the cost pressure, especially on European automotive manufacturers, is increasing, requiring every component in the vehicle to be developed and produced in a cost-optimized manner. Both aspects, as well as the environmental requirements for control units in automotive systems, limit the selection of suitable microcontrollers. It is becoming increasingly important to optimally utilize the limited resources of the available hardware in order to address technical and regulatory challenges in the future. This work examines currently available alternatives to Trusted Execution Environments for use in real-time capable automotive systems.

In addition, this work examines emerging vehicle requirements that relate directly to the functional responsibilities of a Trusted Execution Environment. In this context, two aspects deserve particular mention: the continuous expansion of functionality in already delivered vehicles (in the context of the Software Defined Vehicle), as well as crypto agility. Following these considerations, a concept will be presented that addresses future challenges for TEEs.

As an alternative approach to ensure that vehicles developed today and produced tomorrow remain safe and secure for years to come, the Firmware Security Module (FSM) is introduced, which implements the same set of security requirements entirely in software and thereby adopts a fundamentally different approach. In this context, substantially greater emphasis is placed on resource scalability, hardware independence, and updateability. These characteristics are of critical importance for future TEEs, as they must be capable of accommodating forthcoming changes in technology, regulatory frameworks, and automotive development priorities.

The limitations of the FSM - such as the lack of a true entropy random number generator - demonstrate that a complete abandonment of hardware-based security primitives is not feasible. Consequently, a hybrid architecture combining the FSM with a conventional TEE represents a viable approach to compensating for these deficits.

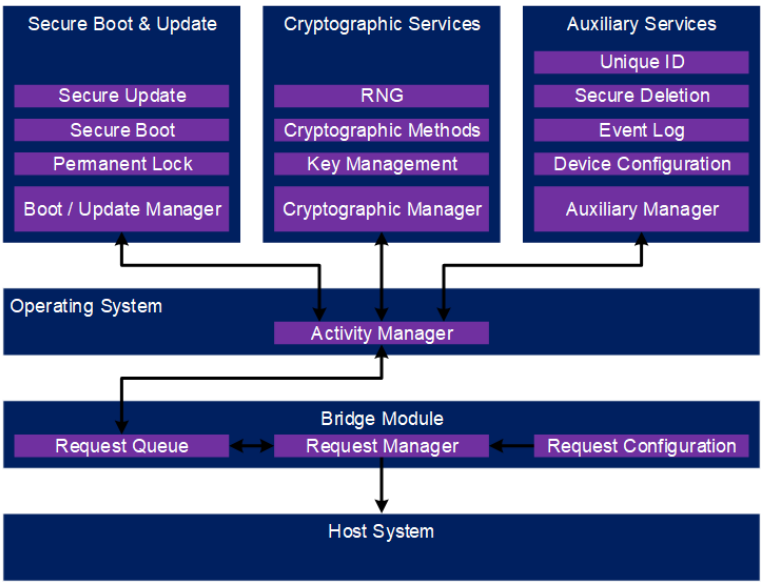

Fig. 1 Basic Structure of the Firmware Security Module