Mitigation of Automotive Control Modules Hardware Replacement-based Attacks Through Hardware Signature

Mitigation of Automotive Control Modules Hardware Replacement-based Attacks Through Hardware Signature

Franco Oberti[1,2], Ernesto Sanchez[1], Alessandro Savino[1], Filippo Parisi[2], and Stefano Di Carlo[1]
[1]*Control and Computer Eng. Dep., Politecnico di Torino* Torino, Italy
[2]*PUNCH Torino S.p.A.*, Torino, Italy

***Abstract*—Authentication of hardware modules connected through Controller Area Networks (CAN) in modern vehicles is becoming an increasing security issue. Untrusted modules introduced on the market may alter the secure boot infrastructure of a complex vehicle, thus completely compromising its security. This paper introduces the problem and highlights a preliminary idea for reaching better protection and preventing or limiting this category of attacks.**

I. Introduction

Nowadays, automotive architectures are supported by different communication systems for connecting the vehicle. One of the most important vehicle networks is the Controller Area Network (CAN) bus. By default, each vehicle has at least a *public* CAN bus reachable through the On-Board Diagnostic (OBD) port. The automotive domain is becoming quite profitable for attackers. From one side, vehicle owners demand the ability to manipulate the system's parts for reaching better performance or bypassing annoying service procedures. On the other side, the use of untrusted parts could drive a no return point for a company's credibility or destabilize an entire market domain. In particular, a third party may devise an attack intended to damage or take advantage of competitors or other market domains, for example, making them easy to steal.

This paper shows a possible attack scenario and introduces preliminary ideas to implement countermeasures.

II. Attack Model

Nowadays, vehicles embed several control modules, whose number depends on the product's market level (e.g., economy, core, premium, luxury). Higher levels require more functionalities, increasing their complexity. Every module has at least one access to a CAN bus since alternatives, such as communication Over-The-Air (OTA), are not currently considered.

A well-designed secure boot [1] is among the most efficient protection against cyber-attacks to real-time embedded modules in automotive. At each bootstrap, the system validates the signature of each memory segment. Moreover, code updates require an authentication mechanism to avoid the injection of potentially counterfeit software. In this scenario, new market leanings might target such security measures, neutralizing boot signatures, and compromising the full system security, granting unauthorized software to run in the system with potential hazards to the safety of the entire vehicle.

Authors contacts: {franco.oberti, alessandro.savino, ernesto.sanchez, stefano.dicarlo}@polito.it and filippo.parisi@punchtorino.com

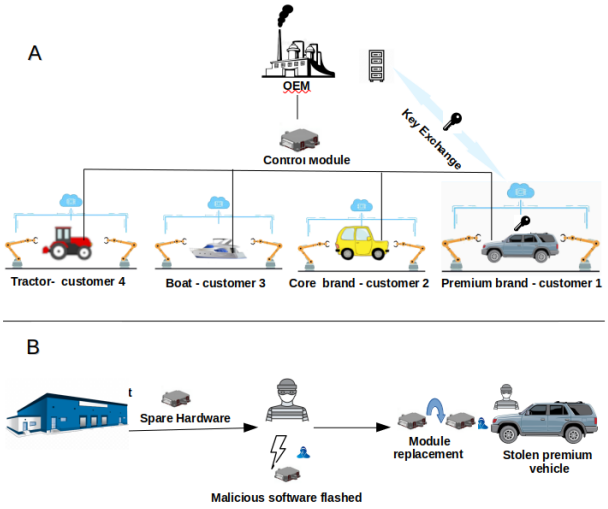

Fig. 1. Attack Model Overview: (a) the same control module is in several application domains and (b) a module can be easily reworked from one domain to another.

The automotive market pushes competition in terms of costs to its limit by exploiting the economy of scale. As depicted in Fig. 1.a, several suppliers provide the same hardware platform to several customers who act in different heterogeneous domains (e.g., automotive, marine, agriculture, general-purpose equipment) with different cyber-security requirements [2].

Non-secure hardware modules can be easily reworked to serve another domain that adopts the same hardware platform (Fig. 1.b). If this domain requires cyber-security (e.g., secure boot), such hardware replacement may bypass the code signature mechanism allowing any software execution. Therefore, hardware platforms must guarantee authenticity.

Physical Unclonable Functions (PUF) and Logic Locking, which are techniques proposed in the literature for hardware fingerprinting, are hard to be exploited in the automotive domain. In vehicles, control modules operate in an extensive range of environmental conditions (i.e., temperature, pressure, humidity) that may severely impact the PUF challenge's success [3], [4]. Moreover, external hardware or other control modules need to validate the challenges, hence defining an

additional custom infrastructure acting in parallel to CAN Bus. Hardware replacement events could also become hard to manage. While PUFs are very powerful in identifying every hardware device, the automotive domain is more interested in tracking control modules associated with a selected customer. Logical locking is a hardware technique based on integrating a locking mechanism into the circuit such that it produces faulty outputs whenever an incorrect key is provided [5], [6], [7]. However, the faulty outputs may generate hazards and violate safety rules in the automotive system.

III. Countermeasure: Hardware Signature for Automotive Secure Modules

Hardware signature based on challenge-response is a viable solution to the scenario introduced in Section II. The idea is to provide each control module board with a custom IC able to generate a compatibility discontinuity among hardware platform subdomains. In case of hardware integrity violation, a recovery action initiated by other controllers can exclude as much as possible non-authentic hardware for keeping the system safe. Certifying subdomains is a way to identify different customers keeping their products separated. This separation prevents using the same subdomain of control modules that target different application domains even if made by the same Original Equipment Manufacturer (OEM). Fig. 2 provides a high-level block diagram of the proposed hardware signature module.

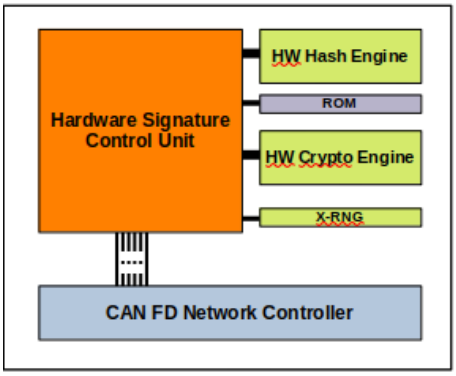

Fig. 2. Hardware Signature for Automotive Control Modules Block Scheme. The module implements basic secure hashing and crypto primitives managed by a control unit, with basic parameter stored in ROM. X-RNG is a random number generator for supporting challenge-response authentication using messages exchanged through CAN FD controller. Eventually, the Control Unit contains all authentication strategies and logic.

At random intervals, modules validate themselves reciprocally with a challenge-response secure scheme. The response to a challenge communicated by a module over the CAN bus must be validated by a certain number of nodes belonging to the network. This validation protocol must be iterated until all devices of the CAN network are validated. In case of a failure, validation shall restart. A threshold on the maximum number of tolerable sequential fails may be used as a parameter to move the system in recovery mode.

The key distribution is based on the already existing infrastructure. Silicon vendors provide ICs with the right company's signature already in ROM. There is a unique global signature for each customer to reduce complexity. The hardware signature ROM is programmed with a reserved value directly by OEM for all customers who do not require a secure platform. If those parts are mounted in a secure vehicle, the challenge-response validation will fail, and the system will shift in a recovery mode. In case the signature secret key of a carmaker is violated, it does not become a threat since all platforms will already have a proper programmed secret key signature in ROM and the spare parts. Led by this proposal, additional costs do not exceed 2% of the overall amount for Engine Control Unit (ECU)'s price. An initial evaluation concept consists of a virtual hardware signature running in emulated automotive system architecture.

IV. Remaining vulnerabilities

While the proposed idea addresses the considered attack models, some minor vulnerabilities remain and need to be addressed.

As for all security devices, the Hardware Signature Module must store secret keys. If an attacker violates the OEM's keys, the entire authentic hardware system is compromised.

The hardware signature IC from a secure device could be desoldered to steal the secret key and then soldered on an equivalent board not targeting secure applications for making it compatible with secure vehicles.

References

[1] R. R.V. and K. A. Secure boot of embedded applications - a review. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 291–298, 2018.

[2] C. Lin and A. Sangiovanni-Vincentelli. Cyber-security for the controller area network (can) communication protocol. In *2012 International Conference on Cyber Security*, pages 1–7, 2012.

[3] H. Kang, Y. Hori, and A. Satoh. Performance evaluation of the first commercial puf-embedded rfid. In *The 1st IEEE Global Conference on Consumer Electronics 2012*, pages 5–8, 2012.

[4] R. Soga and H. Kang. Physical unclonable function using carbon resistor. In *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, pages 559–561, 2020.

[5] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri. On improving the security of logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(9):1411–1424, 2016.

[6] K. Juretus and I. Savidis. Increased output corruption and structural attack resilience for sat attack secure logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(1):38–51, 2021.

[7] T. Thangam, G. Gayathri, and T. Madhubala. A novel logic locking technique for hardware security. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, pages 1–7, 2017.

[8] F. Stellari, P. Song, A. J. Weger, J. Culp, A. Herbert, and D. Pfeiffer. Verification of untrusted chips using trusted layout and emission measurements. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 19–24, 2014.

[9] M. Majzoobi and F. Koushanfar. Time-bounded authentication of fpgas. *IEEE Transactions on Information Forensics and Security*, 6(3):1123–1135, 2011.

[10] M. B. Bahador, M. Abadi, and A. Tajoddin. Hpcmalhunter: Behavioral malware detection using hardware performance counters and singular value decomposition. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 703–708, 2014.