An experimental model for in-vehicle networks and subsystems

Bogdan Groza[1], Horatiu Gurban [1] and Pal-Stefan Murvay[1]
[1] *Department of Automatics and Applied Informatics, Politehnica University of Timisoara, Romania*
{*bogdan.groza, horatiu.gurban, pal-stefan.murvay*}*@aut.upt.ro*

Keywords: in-vehicle networks, automotive security, embedded devices

Abstract: We pursue an experimental setup that gathers various in-vehicle networks and subsystems that are critical from a security perspective. As cyber-attacks to cars have become a reality, the model comes handy for both research and engineering education. The usefulness of this empirical model stems from both being helpful in creating a realistic view on the security of automotive systems and for creating security awareness. We do congregate in our setup various communication buses, e.g., CAN, LIN and FlexRay, and bring connectivity between several low and high-end automotive-grade development boards that are linked to off-the-shelf in-vehicle components, e.g., an instrument cluster and an infotainment unit, etc. The setup serves as a concise and practical representation of in-vehicle subsystems, network topologies and highlights security implications.

1 Motivation

Contemporary cars are the result of an evolution that spans over more than a century. While most of the electronic technologies inside cars were designed in the past few decades, there is a high degree of heterogeneity in modern vehicles in terms of both existing subsystems and networks. Recently, a new challenge has emerged for vehicular technologies: designing security solutions to withstand adversarial threats similar to what already exists in the world of modern computers.

In the light of the recently reported attacks on in-vehicle networks (Checkoway et al., 2011), (Koscher et al., 2010), as both academic researchers and the industry are working on designing solutions to threats that have not been envisioned in the past, it becomes necessary to bring experimental models both for researchers and for engineers in order to make them familiar with existing targets and attack surfaces inside a car. In this work we target the design of an experimental model that can be used for both testing security solutions and creating awareness.

The security community has been constantly aware that security by itself is not a product but a process. Consequently it is not possible to design security solutions disconnected from the system were they finally reside. The situation is not at all distinct when devising security for in-vehicle networks or components. Designing security solutions can clearly benefit from an experimental setup that offers a general view of in-vehicle buses and subsystems. This assists a more realistic perspective in designing such solutions and offers a potential testbed where solutions can be put in practice. We portray the experimental setup that we design in Figure 1 and precise details on the components will be given in a forthcoming section.

We structure our presentation as follows. In sections 2 we discuss the attack surfaces and targets in terms of in-vehicle buses and vehicular subsystems that are common targets for such attacks and relevant for our model along with some useful tools for attacking in-vehicle networks. In section 3 we discuss existing security solutions while section 4 gives details on the experimental setup. Section 5 holds the conclusion of our work.

2 Networks and subsystems

We give a brief description of communication buses, subsystems and attack tools that are target of our experimental setup.

2.1 Communication buses

We enumerate the most relevant communication buses and interfaces in the automotive industry:

1. the CAN bus (Controller Area Network) is one of the first buses adopted by the automotive industry and still largely used in this area. It is currently

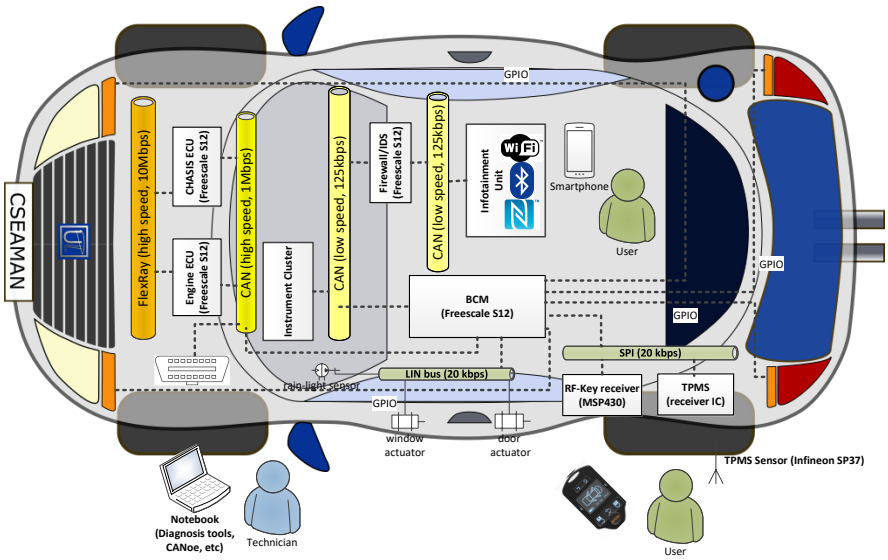

Figure 1: The experimental setup targeted by our work

available in three forms: low-speed fault tolerant CAN (up to 125 kbps, defined in ISO11898-3), high-speed CAN (up to 1Mbit ISO11898-2) and the more recently proposed CAN-FD (CAN with Flexible Data-Rate) which improves the transmission rate allowing bit-rates of up to 5 Mbps. In addition, CAN-FD increases the maximum payload of a message to 64 bytes instead of 8 in the classical CAN.

2. FlexRay is an automotive bus designed to fulfill the communication requirements of X-by-Wire Systems. It is mostly intended for scenarios where high bit-rates are required (up to 2 channels at 10Mbps) and uses TDMA(Time Division Multiple Access) as access method (which assures fixed communication latency). Due to the higher hardware cost and the increased complexity of the required software drivers, FlexRay may fail to entirely replace CAN but it is generally used to interconnect safety-critical ECUs.
3. LIN (Local Interconnect Network) is a serial bus designed to complement CAN networks, offering a cost efficient solution for connecting sensors and actuators in small networks where the maximum speed cannot exceed 20kbps. It is generally used for peripherals.
4. Wireless transmission standards from the consumer electronic market such as WiFi and Bluetooth are now being available in both low and high-end vehicles. This comes with benefits for the driver and passengers by allowing the interaction between the vehicle and smart electronic devices, e.g., phones, tablets and watches, and definitely adds new challenges from the security perspective.

2.2 In-vehicle subsystems

The organization of in-vehicle networks in subsystems dedicated to the main functional domains came as a natural development with the increasing demand for complexity of electronics inside cars. Most commonly, in-vehicle network architectures include the body, infotainment and telematics, power-train, chassis and diagnostics as main subsystems. Other more complex topologies consider additional domains such as driver assistance and safety while, when not specifically nominated, these functionalities are assimilated by the other subsystems. Each of these subsystems is responsible for a set of specific functionalities which can be targeted by attacks. In what follows, we present several common functionalities in relation to the subsystems of which they are a part of and included in our model:

1. Body related subsystems integrate all functionalities that are not related to the vehicle dynamics but are required as basic vehicle functionalities, e.g., wind-shield wipers, lighting, window lift, air conditioning, door locks, immobilizer, etc. Often several of these functionalities, if not all, are integrated in a single module called the Body Control Module (BCM). Successful attacks on the body subsystem can have various consequences. They might lead to the theft of the vehicle in case the

immobilizer and the locking systems are targeted while attacks that affect the exterior lighting during night driving can potentially lead to life threatening situations.

2. Infotainment and telematics handles information services and entertainment aspects intended for enhancing user experience. Infotainment modules can be found in virtually all modern vehicles as all are equipped with instrument clusters and at least a basic multimedia system. The telematic domain provides functionalities that offer support for communication between the vehicle and the exterior world: GPS, radio, mobile telecommunication, etc. While tampering with infotainment or telematic systems may appear to have no serious consequences, recently reported attacks (Checkoway et al., 2011) prove that these systems can provide means to infiltrate the vehicle and gain access to its internal bus providing opportunities for attacks on other subsystems.

3. Power-train systems are responsible with the longitudinal propulsion of the vehicle and include components such as the engine, transmission and related subsystems like the fuel injection, emission control and gear shift. Given the critical nature of the control systems involved in the power-train domain, attacks aimed at this system can result in serious traffic incidents. The real-time nature of power-train related subsystems makes them susceptible to DoS attacks by generating high bus-loads that prevents timely processing.

4. Chasis is the subsystem which sums up components that control the wheels and their movement. Steering and braking are some of the basic functionalities handled by the chassis subsystem, but it is also responsible with more advanced functions such as the anti-lock braking, electronic stability control, etc. It is clear that performing malicious actions on this category of systems can have severe effects.

5. The diagnostics system was introduced as a necessity due to the complexity of modern vehicles. This system, which has become mandatory in newly produced vehicles in the US and the EU, is responsible with self-diagnosis and reporting functionalities. Therefore, all vehicles must provide a physical port for connecting an external diagnosis tool to monitor system variables and read-out existing trouble codes. This port is know as the OBD (On-Board Diagnosis). The vehicle diagnosis system is important from a security perspective as it was previously used as an entry-point for many of the attacks reported so far.

2.3 Attack instruments: CAN sniffers

Whether one wants to mount attacks on in-vehicle networks or build secure communication protocols, a set of hardware and software tools is required for tapping into the communication buses. Most of the currently reported attacks use the CAN bus since it is the most widely used in automotive networks.

A wide range of off-the-shelf PC to CAN interfacing devices are available, at affordable prices, for general purpose applications. Some products come with accompanying applications that offer an out-of-the-box support for CAN sniffing using a PC. Generic solutions exist for those PC to CAN devices that do not come with appropriate software for network protocol analysis, such is the case of the well known Wireshark which can be configured to analyse CAN packets by using SocketCAN. SocketCAN(Smith, 2016) implements CAN drivers and networking stack for different CAN transceivers and uses the CAN bus like a generic network interface. Kayak [1] is a Java application for CAN bus diagnosis and monitoring. It uses Socketcand Daemon which allows the usage of SocketCAN with programing languages that are not able to access the the directly the low-level CAN sockets.

Tools for CAN attacks can be built with little effort using readily available components for embedded hobbyists, e.g. an attack on a Mini Cooper instrument cluster using an Arduino based setup is demonstrated in (Staggs, 2013). EcomCAT software was developed and used by Valasek and Miller in (Miller and Valasek, 2013). It is a CAN analysis tool for monitoring the CAN network and for injecting messages by using ECOM cables.

More complex solutions are available in the form of professional-grade devices and software tools provided by suppliers of automotive industrial development and testing solutions such as Vector and National Instruments. However, the cost of such solutions is much higher and may not prove to be reasonable for some basic sniffing and replay attacks.

Vector CANoe [2] is a well known automotive software used for development and testing of ECUs. It can simulate but also interface with all the communications standards that are available today in the automotive industry.

One important aspect is the definition of CAN network proprieties, of the ECUs that are connected to the bus and of the CAN frames, associated signals and attributes. In the automotive industry the most used format is implemented by Vector .dbc files which can be created and updated by using the Vector CANdb++

[1] https://github.com/dschanoeh/Kayak

[2] https://vector.com/vi_canoe_en.html

software. This proprietary file format is not documented but an open source alternative exists ,CANBabel [3] , which converts the .dbc files in a XML based format .kdx. This file format can be used with the Kayak software, the XML format facilitating the creation and the update of this file.

3 Security solutions: cryptographic protocols

Existing security solutions or academic research works that have not yet reached practical implementations can be discussed in connection with the buses and components that are part of our model.

3.1 Security solutions for in-vehicle buses

We point out several proposed security solutions for the communication buses and interfaces that are present in our experimental model:

1. The CAN bus is likely the most explored in-vehicle bus in terms of security solutions. Existing research works span from the use of basic message authentication codes (Hartkopp et al., 2012), TESLA like protocols (Groza and Murvay, 2013), group key sharing (Groza et al., 2012), physical layer oriented key allocation, security aware signal allocation in frames (Lin et al., 2015), hardware components that discard forged frames (Kurachi et al., 2014), etc.
2. FlexRay received little attention compared to the CAN bus (the main reason is likely that FlexRay capable ECUs are more expensive and harder to use for academic research). Due to its time triggered nature, TESLA like protocols appear to be more appropriate and these have already been explored in (Han et al., 2014), (Gu et al., 2016).
3. LIN is the most limited bus in terms of bandwidth, a reason for which it is unlikely for security solution to be ever devised for this bus. Due to its use in controlling vehicle body components that are in turn controlled by other ECUs from remote, e.g., over the CAN bus, it is less likely for this bus to require any security mechanisms.
4. RF communication interfaces are employed for vehicle keys and tire sensors. Existing security protocols or cryptographic primitives that are proprietary proved to be flawed on careful analysis. For example RF vehicle keys based on Hitag2 (one of the most widely spread solutions) were shown to be flawed in (Verdult et al., 2012). There are many other reported attacks and solutions which we omit here for brevity.

The following wireless interfaces were not specifically designed for vehicles but as they become a constant presence inside cars, they can be exploited as an attack surface:

1. WiFi 802.11 comes with the existing WPA2 security suite which provides a reasonable level of security. Still, since WPA2 security generally relies on passwords, special care should be taken with the authentication credentials.
2. Bluetooth used to provide no security layers at all, but as of version 4.0 support for AES encryption exists. Still a lot of user applications do not require passwords for BT connectivity or they rely on PIN codes that are easy to break. This happens to be the case for BT capable OBD diagnosis tools which are cheap and quite popular today.
3. NFC cards support standard cryptographic functions and can be used for enhancing the security of access control including for in-vehicle immobilizers (Busold et al., 2013).

3.2 Security solutions for in-vehicle subsystems

We now discuss some security solutions along the following subsystems from our model:

1. Electronic immobilizers were designed to protect the theft of the car (Lemke et al., 2006). While simple challenge-response authentication protocols can be used for this purpose, the current trend is to replace mechanical keys with smart electronic devices, e.g., smart-phones. This opens room for new security designs which are discussed by several research works (Busold et al., 2013).
2. Infotainment units are currently equipped with modern Operating Systems (OS). Android appears to be the best candidate OS for such units and is already available in genuine infotainment units or after-market units from third parties. As a Linux derivate, Android comes with several immediate security benefits such as the restricted root access which limits the effects of compromised applications on the system. But Android is not bullet-proof and an Android unit may have been rooted and compromised without driver's knowledge thus opening gate for attacks.

[3] https://github.com/julietkilo/CANBabel

3. The security of the Tire Pressure Monitoring Systems (TPMS) has been addressed by at least two security research works (Xu et al., 2013) and (Solomon and Groza, 2015) since the first attacks were mounted in (Ishtiaq Roufa et al., 2010). Devising security solutions for this system is particularly hard due to the low computational capabilities on existing sensors (e.g., 8-bit processors) and due to energy constraints (sensors rely on batteries).
4. Vehicle instrument clusters have been a constant target of attacks since they used to store the mileage of the vehicle which is a key factor in the reselling value of the asset. Currently, this value is stored in several ECUs inside the car and compromising the instrument cluster alone is ineffective for most modern cars. Work on instrument cluster security is mostly absent.

4 Model components

The experimental setup illustrated in Figure 1 gathers the previously described networks and subsystems. Technical details on the components that we add are available in Table 1. We now discuss network topology and components.

Due to the demand for low latencies, the Engine and Chasis ECUs are connected via a high-speed FlexRay bus. Freescale/NXP MC9S12XF512 are target ECUs for such tasks. Further, both these ECUs are connected to the BCM module over a high speed CAN bus. The same bus can be used for external diagnosis purposes. The BCM module can be implemented on a lower-class MC9S12DG128 or the same MC9S12XF512. The BCM module is responsible for the following body-related functionalities: rain and light sensors on the windshield, door and window actuators that are connected via the LIN bus, and the RF-key and TPMS receiver for which communication is done over a serial port. For implementing RF functionalities in TPMS sensors and key we plan to use low-cost, RF-capable controllers such as Texas Instruments MSP430F2274. The front and rear lights are controlled over General-purpose input/output (GPIO) ports. Further, the BCM is connected on a low speed CAN to the instrument cluster (dashboard) and to the Infotainment unit. For the instrument cluster we use an off-the-shelf cluster.

The infotainment unit as a potential third-party (after-market) device is connected to the CAN bus but a firewall prevents potential attacks. A low cost ECU as the MC9S12DG128 can be used for this purpose. A firewall is needed since the Infotainment units require CAN bus connectivity (e.g., for controlling the unit via steering-wheel buttons) but is clear that such a unit once plugged to the CAN bus can take control and inject messages at will. The firewall prevents this from happening. It is an open question if the OBD diagnosis should be placed behind the same firewall or technicians are trustworthy individuals. For the Infotainment unit we plan to use off-the-shelf units from third party vendors, e.g., Erisin ES2508B is one such example.

Three actors are also present on the model: one technician and two users. Their presence only suggests some use-cases for which security should be considered: technicians that have access over the OBD port, regular user access to the car and connectivity to the infotainment unit.

Table 1: Characteristics of the experimental model devices

Prod.	Unit characteristics	Connectivity
NXP	Target ECU: Firewall/IDS MC9S12DG128, 16bit, 50MHz Flash: 128KB, EEPROM: 2 KB SRAM: 8KB	LIN: 1 ch. CAN: 2 ch. IIC, 2xSCI, 2xSPI SAE J1850
NXP	Target ECU: Chasis, Engine MC9S12XF512, 16bit, 50MHz Flash: 512KB, EEPROM: 2 KB SRAM: 32KB	FlexRay: 2 ch. CAN: 1 ch. IIC, 2xSCI, 2xSPI
TI	Target ECU: RF-Key receiver MSP430F2274, 16bit, 16MHz Flash: 32KB, RAM: 1KB	GPIO: 32 IIC, UART, SPI
Erisin	Infotainment Unit: Erisin ES2508B Rockchip PK3188 Quad-Core Cortex A9, 1.6GHz 1GB RAM, 0.98GB ROM Android 4.4.4	WiFi 802.11b/g/n Bluetooth CAN, USB, GPS

5 Conclusion

Cars are complex ecosystems, an experimental model that covers about a dozen in-vehicle subsystems and buses does not offer a complete view but gives useful hints in assessing the vehicle as a system. Our model groups the most significant in-vehicle buses: CAN, LIN and FlexRay and also some wireless interfaces: BlueTooth, WiFi and RF. It also addresses some relevant subsystems: the instrument cluster, infotainment units, tire sensors as well as several controllers for BCM, engine and chasis functionalities. The challenge in designing security solutions

stems from the fact that security must be transparent to the user, ideally requiring no interaction with the user and small (almost inexistent) delays or busloads. Any security mechanisms that requires more interaction comes at the cost of usability while performance overhead may compromise overall system performance. This experimental model may provide a more realistic setup for validating solutions in this regard.

Acknowledgement. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project number PN-II-RU-TE-2014-4-1501 (2015-2017).

REFERENCES

Busold, C., Taha, A., Wachsmann, C., Dmitrienko, A., Seudié, H., Sobhani, M., and Sadeghi, A.-R. (2013). Smart keys for cyber-cars: Secure smartphone-based NFC-enabled car immobilizer. In *Conference on Data and Application Security and Privacy*, pages 233–242. ACM.

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T., et al. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*.

Groza, B., Murvay, P.-S., Van Herrewege, A., and Verbauwhede, I. (2012). LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks. In *11th International Conference on Cryptology and Network Security, CANS 2012, Springer-Verlag, LNCS*.

Groza, B. and Murvay, S. (2013). Efficient protocols for secure broadcast in controller area networks. *IEEE Transactions on Industrial Informatics*, 9(4):2034–2042.

Gu, Z., Han, G., Zeng, H., and Zhao, Q. (2016). Security-aware mapping and scheduling with hardware co-processors for flexray-based distributed embedded systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):3044–3057.

Han, G., Zeng, H., Li, Y., and Dou, W. (2014). Safe: Security-aware flexray scheduling engine. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1–4. IEEE.

Hartkopp, O., Reuber, C., and Schilling, R. (2012). MaCAN-message authenticated CAN. In *10th Int. Conf. on Embedded Security in Cars (ESCAR 2012)*.

Ishtiaq Roufa, R. M., Mustafaa, H., Travis Taylora, S. O., Xua, W., Gruteserb, M., Trappeb, W., and Seskarb, I. (2010). Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *19th USENIX Security Symposium*, pages 11–13.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., et al. (2010). Experimental security analysis of a modern automobile. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 447–462. IEEE.

Kurachi, R., Matsubara, Y., Takada, H., Adachi, N., Miyashita, Y., and Horihata, S. (2014). CaCAN - centralized authentication system in CAN (controller area network). In *14th Int. Conf. on Embedded Security in Cars (ESCAR 2014)*.

Lemke, K., Sadeghi, A.-R., and Stüble, C. (2006). Anti-theft protection: Electronic immobilizers. In *Embedded Security in Cars*, pages 51–67. Springer.

Lin, C.-W., Zhu, Q., and Sangiovanni-Vincentelli, A. (2015). Security-aware modeling and efficient mapping for CAN-based real-time distributed automotive systems. *IEEE Embedded Systems Letters*, 7(1):11–14.

Miller, C. and Valasek, C. (2013). Adventures in automotive networks and control units. *DEF CON*, 21:260–264.

Smith, C. (2016). *The Car Hacker's Handbook*. No Starch Press.

Solomon, C. and Groza, B. (2015). LiMon - lightweight authentication for tire pressure monitoring sensors. In *1st Workshop on the Security of Cyber-Physical Systems*.

Staggs, J. (2013). How to hack your mini cooper: reverse engineering can messages on passenger automobiles. *Institute for Information Security*.

Verdult, R., Garcia, F. D., and Balasch, J. (2012). Gone in 360 seconds: Hijacking with hitag2. In *Proceedings of the 21st USENIX conference on Security symposium*, pages 37–37. USENIX Association.

Xu, M., Xu, W., Walker, J., and Moore, B. (2013). Lightweight secure communication protocols for in-vehicle sensor networks. In *Workshop on Security, privacy & dependability for cyber vehicles*, pages 19–30. ACM.