A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay

Dennis K. Nilsson[1], Ulf E. Larson[1], Francesco Picasso[2], and Erland Jonsson[1]

[1] Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
[2] Department of Computer Science and Engineering
University of Genoa
Genoa, Italy
{dennis.nilsson,ulf.larson,erland.jonsson}@chalmers.se,
francesco.picasso@unige.it

Abstract. The automotive industry has over the last decade gradually replaced mechanical parts with electronics and software solutions. Modern vehicles contain a number of *electronic control units* (ECUs), which are connected in an in-vehicle network and provide various vehicle functionalities. The next generation automotive network communications protocol FlexRay has been developed to meet the future demands of automotive networking and can replace the existing CAN protocol. Moreover, the upcoming trend of ubiquitous vehicle communication in terms of vehicle-to-vehicle and vehicle-to-infrastructure communication introduces an entry point to the previously isolated in-vehicle network. Consequently, the in-vehicle network is exposed to a whole new range of threats known as cyber attacks. In this paper, we have analyzed the FlexRay protocol specification and evaluated the ability of the FlexRay protocol to withstand cyber attacks. We have simulated a set of plausible attacks targeting the ECUs on a FlexRay bus. From the results, we conclude that the FlexRay protocol lacks sufficient protection against the executed attacks, and we therefore argue that future versions of the specification should include security protection.

Keywords: Automotive, vehicle, FlexRay, security, attacks, simulation.

1 Introduction

Imagine a vehicle accelerating and exceeding a certain velocity. At this point, the airbag suddenly triggers rendering the driver unable to maneuver the vehicle. This event leads to the vehicle crashing, leaving the driver seriously wounded. One might ask if this event was caused by a software malfunction or a hardware fault. In the near future, one might even ask if the event was the result of a deliberate *cyber attack* on the vehicle.

In the last decade electronics and firmware have replaced mechanical components in vehicles at an unprecedented rate. Modern vehicles contain an in-vehicle network consisting of a number of *electronic control units* (ECUs) responsible

E. Corchado et al. (Eds.): CISIS 2008, ASC 53, pp. 84–91, 2009.
springerlink.com

for the functionality in the vehicle. As a result, the ECUs constitute a likely target for cyber attackers.

An emerging trend for automotive manufacturers is to create an infrastructure for performing remote diagnostics and *firmware updates over the air* (FOTA) [1]. There are several benefits with this approach. It involves a minimum of customer inconvenience since there exists no need for the customer to bring the vehicle to a service station for a firmware update. In addition, it allows faster updates; it is possible to update the firmware as soon as it is released. Furthermore, this approach reduces the lead time from fault to action since it is possible to analyze errors and identify causes using diagnostics before a vehicle arrives at a service station.

However, the future infrastructure allows external communication to interact with the in-vehicle network, which introduces a number of security risks. The previously isolated in-vehicle network is thus exposed to a whole new type of attacks, collectively known as cyber attacks. Since the ECUs connected to the FlexRay bus are used for providing control and maneuverability in the vehicle, this bus is a likely target for attackers. We have analyzed what an attacker can do once access to the bus is achieved. The main contributions of this paper are as follows.

- We have analyzed the FlexRay protocol specification with respect to desired security properties and found that functionalities to achieve these properties are missing.
- We have identified the actions an attacker can take in a FlexRay network as a result of the lack of security protection.
- We have successfully implemented and simulated the previously identified attack actions in the CANoe simulator environment.
- We discuss the potential safety effects of these attacks and emphasize the need for future security protection in in-vehicle networks.

2 Related Work

Much research on the in-vehicle network has been on safety issues. Little research has been done on the security aspects in such networks. Only a few papers focusing on the security of those networks have been published. However, the majority of these papers have focused on the CAN protocol. These papers are described in more detail as follows.

Wolf et al. [2] present several weaknesses in the CAN and FlexRay bus protocols. Weaknesses include confidentiality and authenticity problems. However, the paper does not give any specific attack examples.

Simulated attacks have been performed on the CAN bus [3, 4, 5] and illustrate the severity of such attacks. Moreover, the notion of vehicle virus is introduced in [5] which describes more advanced attacks on the CAN bus. As a result, the safety of the driver can be affected.

The Electronic Architecture and System Engineering for Integrated Safety Systems (EASIS) project has done work in embedded security for safety applications [6]. The work dicusses the need for safe and reliable communication for

external and internal vehicle communication. A security manager responsible for crypto operations and authentication management is described.

In this paper, we focus on identifying possible attacker actions on the FlexRay bus and discussing the safety effects of such security threats.

3 Background

Traditionally, in-vehicle networks are designed to meet *reliability* requirements. As such, they primarily address failures caused by non-malicious and inadvertent flaws, which are produced by chance or by component malfunction. Protection is realized by fault-tolerance mechanisms, such as redundancy, replication, and diversity. Since the in-vehicle network has been isolated, protection against intelligent attackers (i.e., *security*) has not been previously considered.

However, recent wireless technology allow for external interaction with the vehicle through remote diagnostics and FOTA. To benefit from the new technology, the in-vehicle network needs to allow wireless communication with external parties, including service stations, business systems and fleet management. Since the network must be open for external access, new threats need to be accounted for and new requirements need to be stated and implemented. Consider for example an attacker using a compromised host in the business network to obtain unauthorized access to the in-vehicle network. Once inside the in-vehicle network, the attacker sends a malicious diagnostic request to trigger the airbag, which in turn could cause injury to the driver and the vehicle to crash.

As illustrated by the example scenario, the safety of the vehicle is strongly linked to the security, and a security breach may well affect the safety of the driver. It is thus reasonable to believe that not only reliability requirements but also security requirements need to be fulfilled to protect the driver.

Since security has yet not been required in the in-vehicle networks, it can be assumed that a set of successful attacks targeting the in-vehicle network should be possible to produce. To assess our assumption, we analyze the FlexRay protocol specification version 2.1 revision A [7]. We then identify a set of security properties for the network, evaluate the correspondence between the properties and the protocol specifications, and develop an attacker model.

3.1 In-Vehicle Network

The in-vehicle network consists of a number of ECUs and buses. The ECUs and buses form networks, and the networks are connected through gateways.

Critical applications, such as the engine management system and the *anti-lock braking system* (ABS) use the CAN bus for communication. To meet future application demands, CAN is gradually being replaced with FlexRay. A wireless gateway connected to the FlexRay bus allows access to external networks such as the Internet. A conceptual model of the in-vehicle network is shown in Fig. 1.

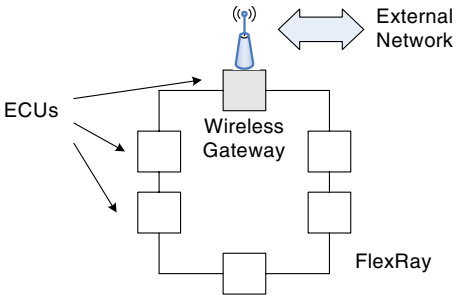

Fig. 1. Conceptual model of the in-vehicle network consisting of a FlexRay network and ECUs including a wireless gateway

3.2 FlexRay Protocol

The FlexRay protocol [7] is designed to meet the requirements of today's automotive industry, including flexible data communications, support for three different topologies (bus, star, and mixed), fault-tolerant operation and higher data rate than previous standards. FlexRay allows both asynchronous transfer mode and real-time data transfer, and operates as a dual-channel system, where each channel delivers a maximum data bit rate of 10 Mbps. The FlexRay protocol belongs to the time-triggered protocol family which is characterized by a continuous communication of all connected nodes via redundant data buses at predefined intervals. It defines the specifics of a data-link layer independent of a physical layer. It focuses on real-time redundant communications. Redundancy is achieved by using two communications channels. Real time is assured by the adoption of a time division multiplexing communication scheme: each ECU has a given time slot to communicate but it cannot decide when since the slots are allocated at design time. Moreover, a redundant masterless protocol is used for synchronizing the ECU clocks by measuring time differences of arriving frames, thus implementing a fault-tolerant procedure. The data-link layer provides basic but strongly reliable communication functionality, and for the protocol to be practically useful, an application layer needs to be implemented on top of the link layer. Such an application layer could be used to assure the desired security properties. In FlexRay, this application layer is missing.

3.3 Desired Security Properties

To evaluate the security of the FlexRay protocol, we use a set of established security properties commonly used for, e.g., sensor networks [8, 9, 10]. We believe that most of the properties are desirable in the vehicle setting due to the many similarities between the network types. The following five properties are considered.

– **Data Confidentiality.** The contents of messages between the ECUs should be kept confidential to avoid unauthorized reads.

- **Data Integrity.** Data integrity is necessary for ensuring that messages have not been modified in transit.
- **Data Availability.** Data availability is necessary to ensure that measured data from ECUs can be accessed at requested times.
- **Data Authentication.** To prevent an attacker from spoofing packets, it is important that the receiver can verify the sender of the packets (authenticity).
- **Data Freshness.** Data freshness ensures that communicated data is recent and that an attacker is not replaying old data.

3.4 Security Evaluation of FlexRay Specification

We perform a security evaluation of the FlexRay protocol based on the presented desired security properties. We inspect the specification and look for functionalities that would address those properties. The FlexRay protocol itself does not provide any security features since it is a pure communication protocol. However, CRC check values are included to provide some form of integrity protection against transmission errors. The time division multiplexing assures availability since the ECUs can communicate during the allocated time slot.

The security properties are at best marginally met by the FlexRay specification. We find some protection of data availability and data integrity, albeit the intention of the protection is for safety reasons. However, the specification does not indicate any assurance of confidentiality, authentication, or freshness of data. Thus, security has not been considered during the design of the FlexRay protocol specification.

3.5 Attacker Model

The evaluation concluded that the five security properties were at best slightly addressed in the FlexRay protocol. Therefore, we can safely make the assumption that a wide set of attacks in the in-vehicle network should be possible to execute. We apply the Nilsson-Larson attacker model [5], where an attacker has access to the in-vehicle network via the wireless gateway and can perform the following actions: read, spoof, drop, modify, flood, steal, and replay. In the following section, we focus on simulating the `read` and `spoof` actions.

4 Cyber Attacks

In this section, we define and discuss two attacker actions: read and spoof. The actions are derived from the security evaluation and the attacker model in the previous section.

4.1 Attacker Actions

Read and spoof can be performed from any ECU in the FlexRay network.

Read. Due to the lack of confidentiality protection, an attacker can read all data sent on the FlexRay bus, and possibly send the data to a remote location via

the wireless gateway. If secret keys, proprietary or private data are sent on the FlexRay bus, an attacker can easily learn that data.

Spoof. An attacker can create and inject messages since there exists no data authentication on the data sent on the FlexRay bus. Therefore, messages can be spoofed and target arbitrary ECUs claiming to be from any ECU. An attacker can easily create and inject diagnostics messages on the FlexRay bus to cause ECUs to perform arbitrary actions.

4.2 Simulation Environment

We have used CANoe version 7.0 from Vector Informatik [11] to simulate the attacks. ECUs for handling the engine, console, brakes, and back light functionalities were connected to a FlexRay bus to simulate a simplified view of an in-vehicle network.

4.3 Simulated Attack Actions

We describe the construction and simulation of the read and spoof attack actions.

Read. Messages sent on the FlexRay bus can be recorded by an attacker who has access to the network. The messages are *Time* stamped, and the channel (*Chn*), identifier (*ID*), direction (*Dir*), data length (*DLC*), and any associated *Data* bytes are recorded. The log entries for a few messages are shown in Table 1. Also, we have included the corresponding message *Name* (interpreted from ID) for each message.

Spoof. An attacker can create and inject a request on the bus to cause an arbitrary effect. For example, an attacker can create and inject a request to light up the brake light. The `BreakLight` message is spoofed with the data value 0x01 and sent on the bus resulting in the brake light being turned on. The result of the attack is shown in Fig. 2. The transmission is in the fifth gear, and the vehicle is accelerating and traveling with a velocity of 113 mph. At the same time, the brake light is turned on, as indicated by the data value 1 of the `BreakLight` message although no brakes are applied (`BrakePressure` has the value 0).

4.4 Effect on Safety Based on Lack of Proper Security Protection

As noted in Section 3, safety and security are tightly coupled in the vehicle setting. From our analysis, it is evident that security-related incidents can affect

Table 1. Log entries with corresponding names and values

Time	Chn	ID	Dir	DLC	Name	Data
15.540518	FR 1A	51	Tx	16	BackLightInfo	128 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15.540551	FR 1A	52	Tx	16	GearBoxInfo	160 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15.549263	FR 1A	13	Tx	16	ABSInfo	59 0 0 0 31 0 0 0 0 0 0 0 0 0 0 0
15.549362	FR 1A	16	Tx	16	BreakControl	0 64 31 0 0 0 0 0 0 0 0 0 0 0 0 0
15.549659	FR 1A	25	Tx	16	EngineData	73 8 81 44 140 10 152 58 0 0 0 0 0 0 0 0
15.549692	FR 1A	26	Tx	16	EngineStatus	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

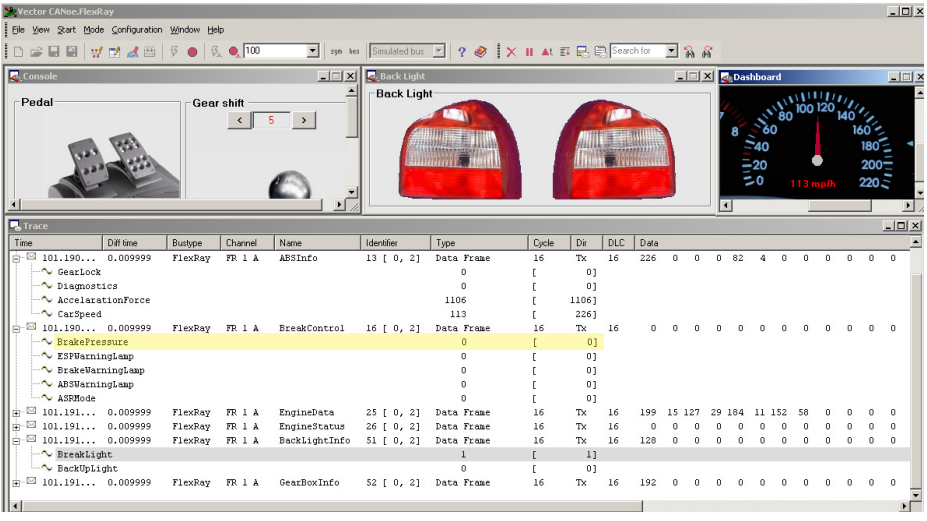

Fig. 2. Spoof attack where the brake lights are lit up while the vehicle is accelerating and no brake pressure is applied

the safety of the driver. Our example with the spoofed brake light message could cause other drivers to react based on the false message. Even worse, spoofed control messages could affect the control and maneuverability of the vehicle and cause serious injury to the drivers, passengers, and other road-users. It is therefore imperative that proper security protection is highly prioritized in future development of the protocol.

5 Conclusion and Future Work

We have created and simulated attacks in the automotive communications protocol FlexRay and shown that such attacks can easily be created. In addition, we have discussed how safety is affected by the lack of proper security protection. These cyber attacks can target control and maneuverability ECUs in the in-vehicle network and lead to serious injury for the driver. The attacker actions are based on weaknesses in the FlexRay 2.1 revision A protocol specification. The security of the in-vehicle network must be taken into serious consideration when external access to these previously isolated networks is introduced.

The next step is to investigate other types of attacks and simulate such attacks on the FlexRay bus. Then, based on the various attacks identified, a set of appropriate solutions for providing security features should be investigated. The most pertinent future work to be further examined are prevention and detection mechanisms. An analysis of how to secure the in-vehicle protocol should be performed, and the possibility to introduce lightweight mechanisms for data integrity and authentication protection should be investigated. Moreover, to

detect attacks in the in-vehicle network a lightweight intrusion detection system needs to be developed.

References

1. Miucic, R., Mahmud, S.M.: Wireless Multicasting for Remote Software Upload in Vehicles with Realistic Vehicle Movement. Technical report, Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202 USA (2005)
2. Wolf, M., Weimerskirch, A., Paar, C.: Security in Automotive Bus Systems. In: Workshop on Embedded IT-Security in Cars, Bochum, Germany (November 2004)
3. Hoppe, T., Dittman, J.: Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In: Proceedings of the 2nd Workshop on Embedded Systems Security (WESS), Salzburg, Austria (2007)
4. Lang, A., Dittman, J., Kiltz, S., Hoppe, T.: Future Perspectives: The car and its IP-address - A potential safety and security risk assessment. In: The 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP), Nuremberg, Germany (2007)
5. Nilsson, D.K., Larson, U.E.: Simulated Attacks on CAN Buses: Vehicle virus. In: Proceedings of the Fifth IASTED Asian Conference on Communication Systems and Networks (ASIACSN) (2008)
6. EASIS. Embedded Security for Integrated Safety Applications (2006), http://www.car-to-car.org/fileadmin/dokumente/pdf/security_2006/sec_06_10_eyeman_easis_security.pdf
7. FlexRay Consortium. FlexRay Communications System Protocol Specification 2.1 Revision A (2005) (Visited August, 2007), http://www.softwareresearch.net/site/teaching/SS2007/ds/FlexRay-ProtocolSpecification_V2.1.revA.pdf
8. Luk, M., Mezzour, G., Perrig, A., Gligor, V.: MiniSec: A secure sensor network communication architecture. In: IPSN 2007: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, pp. 479–488. ACM Press, New York (2007)
9. Perrig, A., Szewczyk, R., Wen, V., Culler, D.E., Tygar, J.D.: SPINS: Security protocols for sensor networks. In: Mobile Computing and Networking, pp. 189–199 (2001)
10. Karlof, C., Sastry, N., Wagner, D.: TinySec: A link layer security architecture for wireless sensor networks. In: SenSys 2004: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, November 2004, pp. 162–175 (2004)
11. Vector Informatik. CANoe and DENoe 7.0 (2007) (Visited December, 2007), http://www.vector-worldwide.com/vi_canoe_en.html