A Survey of Security Threats and Protection Mechanisms in Embedded Automotive Networks

Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaâniche, Youssef Laarouchi

Survey on Security Threats and Protection Mechanisms in Embedded Automotive Networks

Ivan Studnia[1], Vincent Nicomette[2,3], Eric Alata[2,3], Yves Deswarte[2,4]
Mohamed Kaâniche[2,4], Youssef Laarouchi[1]
[1]Renault S.A.S., 1 Avenue du Golf, F-78288 Guyancourt, France
[2]CNRS, LAAS, 7 Avenue du colonel Roche, F-31400 Toulouse, France
[3]Univ. Toulouse, INSA, LAAS, F-31400 Toulouse, France
[4]Univ. Toulouse, LAAS, F-31400 Toulouse, France
Email: {ivan.studnia, youssef.laarouchi}@renault.com, {deswarte,nicomett,kaaniche,ealata}@laas.fr

***Abstract*—Embedded electronic components, so-called ECU (Electronic Controls Units), are nowadays a prominent part of a car's architecture. These ECUs, monitoring and controlling the different subsystems of a car, are interconnected through several gateways and compose the global internal network of the car. Moreover, modern cars are now able to communicate with other devices through wired or wireless interfaces such as USB, Bluetooth, WiFi or even 3G. Such interfaces may expose the internal network to the outside world and can be seen as entry points for cyber attacks. In this paper, we present a survey on security threats and protection mechanisms in embedded automotive networks. After introducing the different protocols being used in the embedded networks of current vehicles, we then analyze the potential threats targeting these networks and describe how the attackers' opportunities can be enhanced by the new communication abilities of modern cars. Finally, we present the security solutions currently being devised to address these problems.**

I. Introduction

The embedding of electronic components into cars is now a well established fact: modern vehicles usually comprise between 30 and 70 ECUs (Electronic Control Units, the embedded computers controlling one or more functions of a vehicle). The amount and complexity of the embedded software are still growing nowadays [1]. These ECUs communicate between them in order to efficiently monitor and control the different vehicular subsystems, therefore forming an automotive network. Like any other computing system, an automotive network can be plagued by vulnerabilities, which can be exploited by an attacker connected to it. However, as the ECUs could not easily be accessed from outside the vehicle, the implementation of security mechanisms into automotive networks was not a major concern until recently.

With today's trends towards interconnections of sensors, actuators and devices, the modern cars now often possess interfaces enabling wired (USB) or wireless (Bluetooth, WiFi, 3G...) communication with the outside world. This trend in the automotive industry will keep increasing with the future deployment of car to car and also car to infrastructure communications.

Therefore, the various computing systems embedded in modern cars can no longer be considered as a closed network, and opportunities of cyber attacks targeting the embedded automotive networks have become a reality.

This paper is organised as follows. In Section II, we first give an overview of the different communication protocols (for both internal and external uses) currently being implemented in today's vehicles. Then in Section III, we analyse the potential threats targeting the automotive network, considering vulnerabilities of a car's internal

networks and attack scenarios deriving from the new communication abilities of a modern car. Section IV is devoted to the survey of the works aiming at implementing security mechanisms in the connected car. Finally, Section V concludes this paper.

II. The automotive network(s)

With so many embedded units, two ECUs that need to exchange data cannot do so through a dedicated point to point connection, because the amount of wire required for a single car would cost too much (and take too much space). Therefore, many ECUs are connected to a bus where any message is broadcast to all the connected nodes. According to the needs, communication between ECUs can use several protocols. A vehicle is therefore made up of several subnetworks interconnected through gateway ECUs. A more comprehensive desciption can for example be found in [2]. We briefly depict a few of them thereafter:

- CAN (Controller Area Network) is a serial bus designed for an automotive use. Data rates go up to 1Mb/s. The medium access protocol is, as in Ethernet networks, based on CSMA/CD (Carrier Sense Multiple Access / Collision Detection) : each node can start emitting if no message is currently being transmitted on the bus. In case of conflict (several nodes trying to emit simultaneously), each ECU applies an arbitration policy by comparing its message identifier with the identifier read on the bus (see table I): the identifier with the highest number of most significant bits set to 0 gains priority. CAN exists in several standards, according to one's needs, and is currently the most used protocol in automotive networks. Many of the studies presented in this survey therefore focus on CAN.
- LIN (Local Interconnect Network) uses a master-slave model, where a master node and up to 16 slave nodes share a bus. A slave can only send a message if previously asked to by the master. Rate can go up to 20kb/s. This protocol is a low-cost solution to connect ECUs that do not need high data rates. It is therefore usually used for controlling a car's comfort elements, such as electric window lifts or windshield wipers.
- FlexRay was conceived by the FlexRay Consortium to be a successor to CAN, offering better rates (up to 10Mb/s). Such rates enable for example X-by-Wire technologies, that is to say the electrical control of currently mechanical control systems like the steering wheel (Steer-by-wire) or the brakes (Brake-by-wire). However, the higher production costs of FlexRay hinder its widespread use.
- MOST (Media Oriented Systems Transport) is used to carry multimedia data into the car via optical fiber. MOST is a synchronous network offering multiple data channels as well as a control channel used to set up which data channels a sender and a receiver will use. Synchronous data channels are used to transfer streaming data (such as audio or video signals) but MOST also implements asynchronous data transfer mechanisms (for example while retrieving data from the Internet) by using some dedicated channels. It offers rates going up to 24Mb/s.
- Finally, the use of Ethernet in the automotive context is being considered in a near future.

The SAE (Society for Automotive Engineers) classified communication protocols in four (formerly three) categories, ranging from A to D, according to their rates and offered features. Details on this classification can be found in table II.

Moreover, modern vehicles can also receive data from external sources, ranging from a USB flash drive plugged into the car media player to online services granted through 3G/4G communications. This trend will be further amplified with the emergence of V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure) communications. This will for example allow a car braking abruptly to alert the following vehicles so that their drivers could react more quickly, or even to automatically trigger an emergency brake.

Table I. STRUCTURE OF A CAN FRAME

SOF	Identifier	Control	Data	CRC	ACK	EOF
1 bit	12/30 bits	6 bits	0 - 64 bits	16 bits	2 bits	7 bits

Table II. SAE CLASSIFICATION OF AUTOMOTIVE NETWORKS

Class	Rate	Use	Examples
A	<10kb/s	Body control	LIN
B	10kb/s →125kb/s	Non critical generic data transfer	CAN-B (Low-speed CAN)
C	125kb/s →1Mb/s	Critical real-time communications	CAN-C (High-speed CAN)
D	>1Mb/s	Multimedia or X-by-wire	MOST, FlexRay

Therefore, a car's internal networks are now complemented by means of communication with external devices. Figure 1 shows a summary of those different external connections.

While all these communication facilities bring new features to the car, they also potentially expose the internal network to the outside world. However, as we will see in part III-B1, even if some protocols implement safety related mechanism, they may be inefficient against attacks relying on a malicious use of the network. While such attacks have been considered unlikely as long as the cars could be considered as a closed network (where any modification therefore implied a prolonged physical access to the car's wiring), the addition of ECUs able to access data from external sources (up to an Internet connection) renders those networks potentially vulnerable to remote computer attacks.

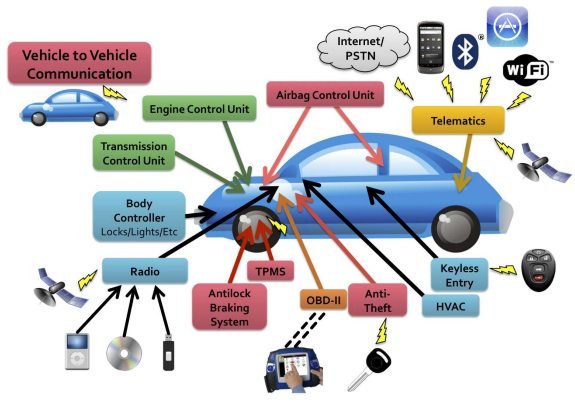

Figure 1. Possible connections of a modern car (from [3])

III. CLASSIFICATION OF ATTACKS

Attacks can be classified according to different criteria, such as those presented in Figure 2. In this section, we consider simply attack goals and attack vectors, whether internal or external.

A. Attack goals

Before looking for potential attack scenarios, one must first identify the possible existing motivations for an attacker to launch an attack against the embedded vehicular network.

Theft: This is perhaps the most obvious motivation at first glance. An attacker could for example exploit a vulnerability present in a wireless communication protocol (cf III-C) to quietly unlock the targeted car and then deactivate the immobilizer or a potential alarm.

Electronic tuning: This case gathers all the situations where the attacker is also the owner of the targeted car. His goal is to make unauthorized modifications in the code or the data contained into one or several ECUs. For example, one could lower the mileage of his car to sell it back at a higher price, do some tuning with the engine settings to gain more power or install unauthorized (unapproved or illegally downloaded) programs into the board computer. Moreover, the car owner can try to bypass a specific authentication mechanism in order to install cheap aftermarket ECUs instead of the more expensive, constructor-approved ones. If the expected consequences of such acts may appear as a minor concern regarding the passengers' health, they can also have possibly unexpected safety (as well as financial in the latter example) implications.

Sabotage: This category regroups all the attacks aimed at deteriorating the vehicle capacities

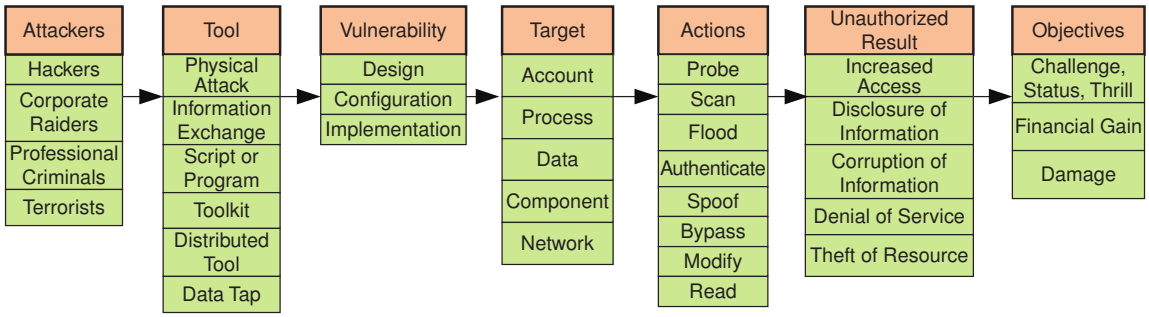

Figure 2. Taxonomy of computer attacks in an automotive context proposed by [4]

through the deactivation of ECUs, the alteration of their software or a denial of service on the network. Consequences range from minor inconveniences (such as a locked air conditioned) to potentially deadly accidents (for example if the brakes are no longer responding). However, a small inconvenience occurring on a few cars could be enough to severely harm a manufacturer's reputation for a long time.

Intellectual property theft: An attacker could try to obtain confidential information about the embedded network of the targeted vehicle. He could do so by eavesdropping on a bus then analysing and identifying the role of each recorded frame or trying to retrieve the source code of an ECU. Such operations could then allow for the production of counterfeited ECUs or the disclosure of new vulnerabilities to potential attackers.

Privacy breach: While the cars embed more and more electronic components, they also store more and more personal information. An attacker could want to retrieve such pieces of information, such as the driver's phone directory and call history, GPS coordinates history or favorite radio frequencies.

Intellectual challenge: Finally, many examples throughout computer science history remind us that we must take into account the attacker solely motivated by the challenge of taking control of a vehicle.

Each of these motivations can be linked to one or more attacker types, each one with different levels of available knowledge, tools and financial resources. Attempts to profile and classify the attackers have been done in [4] and [5], where models of attack scenarios have been defined. We reproduce in Figure 2 the model proposed by [4], derived from the attack taxonomy used by the CERT [6], and adapted to an automotive context. Moreover, in order to gain more knowledge about the attackers, Verendel et al. [7] considered the deployment of honeypots in in-vehicle networks, gathering attack data as the car is moving.

B. Internal attacks

As previously mentioned, some safety-related mechanisms are implemented in current automotive networks, but now security has become a significant concern since malicious actions can have serious consequences on car safety, and these malicious actions are made easier by the recent evolution of car networks.

1) Vulnerabilities on the bus: First, analyses of the buses [8], [9] did highlight vulnerabilities in the current network protocols, with a particular focus on CAN. Even if some ECUs, such as the immobilizer, are secured by specific security mechanisms (such as device authentication), it was shown that CAN cannot by itself guarantee the following security properties:

- Confidentiality: By design, every message sent on CAN is (physically and logically) broadcast to every node. Therefore, a malicious node can easily eavesdrop on the bus and read the content of every frame.
- Authenticity: A CAN frame (see Table I) does not include a field to authenticate its

sender. Therefore, any node can potentially send messages that should only be sent by some other nodes.

- Availability: Arbitration rules in CAN (see II) make it easy for an attacker to cause a denial of service on the bus. For example, an ECU can flood it with high priority frames, therefore forcing every other ECU to stop their transmissions.
- Integrity: CAN uses a CRC to check if a message has been modified by a transmission error, but this is inefficient to prevent an attacker from modifying a correct message or creating a false message, since it is easy to forge a correct CRC for a fake message.
- Non repudiation: There is currently no way for a correct ECU to prove that it has not sent or received a given message.

Since current car networks are unable to guarantee these security properties, we need to analyze if the corresponding vulnerabilities can be exploited.

2) Local attacks: In this section, we describe some documented attacks performed by directly sending packets on an embedded bus (usually a CAN bus). This can be done by plugging an additional device on the targeted bus or through the OBD (On Board Diagnostics) port. OBD refers to a vehicle's ability to identify and report existing problems in its infrastructure. Implementations of OBD are mandatory in every vehicle sold in the US (since 1996) and the European Union (since 2001 for gasoline-powered vehicles and 2004 for diesel-powered ones). These include a standardized communication port used to retrieve diagnostic data generated by the vehicle's sensors. OBD dongles, used to interface a computer with the OBD port of a vehicle, can be legally bought by anyone. For an attacker, this port can be seen as a plug-in entry point into the CAN bus.

In particular, one can use the OBD port to eavesdrop on the bus traffic, but also to send frames. Many documented examples of attacks based on a direct access to the internal network are now available. First, there are examples of attacks from a black box perspective [9] in order to learn the meaning and effects of the frames identifiers and payloads (the protocols are standardized but the contents and effects of the frames depend on the manufacturer, or even on the model). Then, a malicious ECU can replay previously recorded frames and thus send control instructions to other ECUs, thus masquerading the legitimate sources of these instructions [10]. Some ECUs can even be updated through the network and reflashed in that way [11], therefore leaving the vehicle in a compromised state after the attacker's intervention. In [12], Nilsson and Larson introduce the concept of an automotive virus which would trigger only when specific conditions are met (such as the transmission of a given frame, door lock in this case, on the bus).

However, the targeted ECUs may not be on the same bus segment as the attacker's entry point. In [11] Koscher et al. were able to solve this problem by previously targeting and reprogramming the ECUs acting as gateways between the buses, effectively enabling an ECU located on a low-speed, non critical CAN bus to send frames on a safety critical, high-speed CAN bus.

Therefore, if an attacker controls only one single node of the network, current architectures and protocols make it possible for him to gain total control possibly over any other ECU of the vehicle. However, one could object that any of the previously described attacks implied that the attacker already had a physical access onto the bus. Such attack scenarios therefore imply a previous security breach where the attacker had been able to open the vehicle and plug a device onto the network. Moreover, depending on the attacker's goals, quicker and easier non electronic ways may exist (for example, cutting the brake wires instead of hacking the corresponding ECUs).

However, with modern cars wireless communication capacities, an attacker may no longer need to get a physical access to the targeted vehicle. Examples of such attacks are presented in the following section.

C. Remote attacks

In 2011, Checkoway et al. [3] were able to remotely reproduce the attacks described in [11] by finding and exploiting vulnerabilities in a car's

communication interfaces (see Figure 1), therefore not requiring any physical access to the embedded network. These attacks have been sorted by range: indirect physical access, short range wireless access and long range wireless access. We now illustrate these results (along with some other remote attack examples) with possible attack scenarios.

1) Indirect access: We here focus on the attacks relying on a compromised third-party device which will later be connected to the car. If a physical connexion to the network is *in fine* required, that step is no longer performed by the attacker.

OBD port: Section III-B showed that the diagnostic port could be used to attack the automotive network. Here, the actual attack is made against the diagnostic device being plugged into the port. In [3], a so-called pass thru device, plugged into the ODB port and remotely controlled by WiFi from a laptop was compromised: vulnerabilities in the communication API enabled to inject a shell code into the device from another computer on the same WiFi network. Then, the pass thru device emitted malicious packets onto the network each time it became plugged into a new car. Worst, the infected pass-thru device could also in turn attack other identical devices sharing the same WiFi network.

CD player: [3] identified two vulnerabilities in the analyzed player. First, the insertion of a CD containing a file with a specific name tricked the player into believing it to be a firmware update, therefore installing new, malicious software. Moreover, another vulnerability on the decoding of WMA files allowed the team to create a playable audio file that caused the player to emit messages on the bus while reading it. If the first attack is less likely to happen, since a car owner would probably not accept to put an unknown disc into its player, malicious music files downloaded on peer-to-peer networks are a more serious threat.

USB port: Several scenarios can be devised. First, cases similar to the previous one are plausible, where the car media player accesses a corrupted file stored on a USB key. Another possibility would be through the connection of a compromised device (like a smartphone or a mp3 player) which would then perform an attack against the ECU it is connected to. If such an attack has not been reported yet, previous examples of attacks against a mobile phone (for example via bluetooth[1] or after the installation of an installation containing a trojan horse) make this a viable scenario.

2) Short range attacks: This category regroups attacks that use short-range wireless communication technologies. The attacks can be direct, if the attacker tries to directly target a car's communication module or indirect if he targets a driver's device that is already able to connect to the car (e.g., a smartphone).

Wireless pairing of mobile devices: Modern vehicles can sometimes be paired with compatible mobile devices. For example, the driver can connect his phone via Bluetooth and use his car's sound system as a hands free kit. However, the implementation of such wireless protocols into the car can be faulty. Exploiting such vulnerabilities can lead to the retrieval of data stored into the communications unit, the ability to eavesdrop on the conversations (be they phonecalls or conversations between the passengers) or even the compromission of the ECU [3] (and therefore the network).

Car-to-car communications: Communications between a vehicle and other vehicles or roadside infrastructures are probably the next big evolution in the domain of road transport. Indeed, in a few years probably, a car will be able to communicate its status to the neighbouring vehicles. This would for example allow a car to alert its driver in case of an imminent danger (emergency braking of a car ahead, incoming vehicles at a crossroad, etc.) or even to automatically adapt to the new conditions. A detailed risk analysis for intervehicular communications can be found in [13]. Among these risks, we can cite the eavesdropping on the communications, the emission of fake data to a vehicle in order to trigger an unappropriate reaction, and of course a potential compromission of the ECU responsible for car-to-car communications.

TPMS: Tire Pressure Monitoring System is composed of a pressure sensor inside the tire that sends its data to a dedicated ECU located on the CAN via a radio frequency emitter. TPMS are now mandatory in the US, in Europe and soon in

[1] http://trifinite.org/trifinite_stuff_bluebug.html

Japan. In [14], attacks against a TPMS allowed the team to eavesdrop on it from up to 40 meters and send spoofed messages to the monitoring ECU, causing it to turn on tire pressure warning lights at inappropriate times.

Wireless unlocking: Many cars now implement a remote unlocking of their doors or alarms. While some encryption is applied to such instructions sent over the air, it can be cracked, or bypassed. For example, documented attacks against KeeLoq, a block cipher used by several manufacturers, can be found in [15] or [16]. Moreover, Passive Keyless Entry and Start (PKES) systems allow the drivers to unlock and start their cars while keeping their keys in their pockets. In [17], a team was able to perform relay attacks on PKES systems of ten different car models. As a result, by placing an antenna close to the key holder (within a 8m radius) and another near the targeted car, they were able to unlock it then start its engine while the keys were actually 50 meters from the car.

As evidenced by our last example, the "short" range of the aforementioned wireless protocols can sometimes be extended through the use of relays or more powerful antennas. For example, an attack carried via bluetooth has reportedly been made from a distance of over one mile[2].

3) Long-range direct attacks: This category regroups attacks carried over long-range wireless communica- tion technologies.

Telephony: Following the discovery of several vulnerabilities in the telematic unit, Checkoway et al. [3] successfully made it execute custom code downloaded through the 3G network, effectively compromising the vehicle.

Web browsing: In the event that a vehicle embeds a web browser, possible exploits similar to those found on traditional computers and mobile devices are to be considered (e.g., buffer overflow, code injection, etc.).

4) Long range indirect attacks: Finally, we describe here attacks that require a long-range transmission channel and also the compromission of an intermediary device.

App store: In a trend similar to what can be found on the smartphones, some car manufacturers already provide, via the firm online store (similar to the Apple Appstore or the Google Play Store), a selection of downloadable applications for the multimedia unit of their cars. A successful attack against the online store, or a program sold on such a store actually containing a trojan horse (such programs have already been found on the Appstore and the Play Store[3]) would have serious large scale consequences.

Side channel triggers: In [3], an hypothetical scenario is devised in which a backdoor is installed into an ECU of a vehicle compromised through any of the previously described attacks. From that moment on, broadcasts of certain signals (for example via RDS[4]) will trigger the execution of a series of instructions in any compromised vehicle in range of these broadcasts. Catastrophic scenarios can be imagined by combining such techniques with a great amount of previously infected vehicles.

IV. Protection mechanisms

As previously seen, cars are now able to communicate via numerous channels, which can become potential entry points into the embedded network for an attacker. If documented examples of more and more advanced attacks appeared during the last few years, countermeasures are also being developed. In this section, we first present the constraints that must be taken into account while designing security solutions for the automotive environment. Then, we describe the techniques currently being developped to enforce security properties in the automotive networks. We begin with the wireless communications protocols and then focus more specifically on internal defense mechanisms. While this survey makes no claim of comprehensiveness (and can be completed by similar works such as [18] or [5]) we however tried to the best of our knowledge to illustrate the different areas of research currently being explored.

[2]http://trifinite.org/trifinite_stuff_lds.html

[3]http://www.securelist.com/en/blog/208193641/Find_and_Call_Leak_and_Spam

[4]Radio Data System, a protocol used to embed some data in FM radio broadcasts

A. Constraints

Even if usual computing security concepts and methods can be adapted to protect a connected car, important differences still remain and impact the design and set up of automotive security mechanisms. Wolf et al. [5] express the following constraints:

Hardware: Most of a car's embedded computers have strong hardware limitations compared to current traditional computers or smartphones. With such limited computing power and memory, these ECUs are not able to perform advanced cryptographic functions allowing for strong encryption. However, the attacker's hardware may not have such limitations, so a too simple ciphering algorithm could easily be cracked and would therefore prove ineffective (and even counterproductive).

Real time: Similarly, due to the limited computational power of an ECU, longer durations are required to run complex instructions. On the other side, automotive software must deal with real-time constraints, in particular the embedded applications must run in a given time to ensure the safety of the vehicle and its passengers. Therefore, any security mechanism must not impact significantly the embedded software performance.

Autonomy: The driver's attention must overall be focused on the driving. Therefore, the protection mechanisms have to be as autonomous as possible and must only require the driver's attention in extreme situations.

Physical constraints: Some ECUs must be able to sustain physical conditions (high temperatures, moisture, shocks…) that would not be encountered by a traditional computing system.

Lifecycle: The lifecycle of a vehicle (about twenty years) is longer than that of a computer. Embedded security systems must therefore be efficient throughout that duration. Therefore, to prevent obsolescence of security mechanisms, it is advisable to design them to allow an easy updating.

Compatibility: Compatibility must be ensured in two aspects. First, in order to reduce the costs, a security architecture should be as compatible as possible with the currently used embedded technologies (retrocompatibility). Moreover, communications with external sources (devices or other vehicles) must not be hindered by the security mechanisms (interoperability). For example, two distinct car models should not be prevented from communicating because their protocols are incompatible.

B. External communications protections

As seen in the previous section, one vulnerability into the management of the communications with an external device may be enough to entirely compromise the vehicle. Therefore, a first step in order to protect the embedded system would be to secure those channels.

Among the attacks described in III-C, many were allowed by poor implementations of the targeted protocols, flaws in the programming of the involved applications (allowing for buffer overflows) or non-compliance with the manufacturer's specifications. Therefore, such attacks could have been theoretically prevented by strict compliance with good programming practices and by following the existing security recommendations about the communication protocols (for example, [19]). However, the complexity of today's embedded systems combined with the fact that ECUs come from different suppliers can make it almost impossible to check for the compliance with all relevant specifications. Therefore, the integration of additional defense mechanisms in order to secure the communications is essential.

The manufacturers are now fully aware of such issues, as evidenced by the recent large-scale projects between industrial and academic partners. For example, European projects such as SEVECOM [20], PRESERVE [21] or EVITA [22] aim at designing secure communication architectures for internal or intervehicular communications. On a different topic, the goal of OVERSEE [23] is to devise a unified, open and secured multimedia interface managing all the communication protocols.

C. Internal protections

Regarding the security of the communications over the CAN bus, several solutions (not mutually exclusive) are being considered. We can classify them into three categories.

- Cryptographic solutions to authenticate or encrypt the packets transmitted on a bus.

- Solutions detecting anomalies occuring in the system.
- Solutions to ensure integrity of the embedded software.

1) Cryptography: As seen in II, any message emitted on CAN is broadcast to all the nodes connected to the bus. Moreover, there is no proper way of authenticating the sender of a message.

In order to overcome these issues, the implementation of cryptographic solutions on the CAN can enable ECU authentication, integrity checks and encryption of the emitted frames, preventing its reading by nodes not possessing the appropriate keys. Such features are for example proposed in the implementations described in [24], [25] or [26].

However, the computation required to perform strong enough encryption or decryption of the messages can be very time and resource consuming, which is an important issue in a real-time system such as a vehicle. This problem can be addressed by using a hardware module entirely dedicated to cryptographic operations in order to free the ECUs computational capacities. EVITA conceived such a device, called the Hardware Security Module (HSM), which exists in three models implementing various security features according to each ECU requirements [27]. [28] gives examples of a secure key exchange protocol and message encryption using the HSM and an ECU dedicated to key management.

2) Anomaly detection: Other works aim at monitoring the data transmitted between ECUs and assert their legitimacy. A simple and more safety-oriented example can be found in [29] where a module detects if the delay between two frames sent by the monitored ECU is too short, in which case the faulty ECU gets muted.

Moreover, [30] proposed a system where, on every bus, each frame identifier is associated to only one ECU. In other words, such frames can only be sent by one particular ECU and therefore cannot legitimately be sent by the others. Then, whenever a message is emitted on the bus, each ECU checks if the frame identifier is one of its own. If it is the case and if the ECU itself is not the actual sender of the frame currently being emitted, it immediately emits a high-priority alert frame to override the illicit emission.

[31] uses a binary tainting tool to mark the data being used by the ECUs as they are processed and sent on the network. It is then possible to track the origin of malicious instructions in the system. However, this solution is currently quite resource-consuming.

Finally, some works are focusing on the deployment of intrusion detection (resp. prevention) systems (IDS, resp. IPS) in a similar fashion to those found in the traditional computing world. These systems can use two detection methods:

- *Signature-based*: An alert is raised whenever a sequence of frames corresponds to a known signature stored in the system database. If a well defined signature base raises very few false positive, regular updates are required to enable the detection of newly discovered attack patterns. The eight sensors given and discussed in [32] monitoring different aspects of the frames being emitted on the CAN (see table III) can provide an example of the kind of rules required to monitor the bus.
- *Anomaly-based*: This approach requires to define models representing all the possible normal behaviors of the monitored system. Then, anomalies are detected whenever the current state of the system deviates too much from the corresponding model. If such systems can theoretically detect previously unknown attacks, the high complexity of an automotive network makes it difficult to design a model precise enough to prevent false negatives while still allowing exceptional but perfectly legitimate situations. For example, [33] defines the notion of entropy on the CAN and tries to detect sudden deviations of said entropy compared to a reference set.

If these examples of intrusion detection systems applied to an automotive context are still early proofs of concept, the idea seems promising. However, as reminded in IV-A, the automotive environment does not have the same constraints than a traditional computing network. For example,

Table III. LIST OF THE SENSORS DEFINED IN [32]

Sensor	Description
Formality	Correct message size, header and field size, field delimiters, checksum, etc.
Location	Message is allowed with respect to dedicated bus system
Range	Compliance of payload in terms of data range
Frequency	Timing behavior of messages is approved
Correlation	Correlation of messages on different bus systems adheres to specification
Protocol	Correct order, start-time, etc. of internal challenge-response protocols
Plausibility	Content of message payload is plausible, no infeasible correlation with previous values
Consistency	Data from redundant sources is consistent

a car may not be able to update its software (and therefore an IDS signature base) as frequently as a computer. Similarly, as the embedded security systems need to be as autonomous as possible, automatic handling of a false positive could trigger an unnecessary intervention or even endanger the passengers safety.

3) ECU software integrity: Finally, means of ensuring that the vehicle's critical software cannot be affected by an attack are also considered.

First, secure validation of an ECU code can be done in a way similar to the secure boot mechanisms [34] implemented in traditionnal computers. The definition of a trusted base in a vehicle can be done through security modules such as EVITA's HSM or a TPM (Trusted Platform Module) [35].

Ensuring integrity of the multimedia ECU is also one of the main goals of OVERSEE, which is accomplished through the use of a hypervisor (XtratuM[5]) in order to isolate critical software (allowed to write on the buses) from non trusted modules such as the external communication interfaces by putting them into distinct virtual machines. Therefore, should an attacker exploit a vulnerability in a wireless communication protocol, he will not be able to compromise the whole ECU and send messages on the bus (if the hypervisor is able to enforce a strict isolation policy).

V. CONCLUSION

In this paper, we have seen that the lack of existing security mechanisms in the current automotive network architectures has become a serious issue with the addition of wireless communication capacities to the modern cars. Indeed, vulnerabilities in the modules handling such wireless protocols can allow an attacker to remotely access the vehicle embedded network and jeopardize the integrity of possibly every ECU on the network. Therefore, the design and implementation of automotive security mechanisms has become a key issue for automotive manufacturers. We then presented several works aiming at enforcing security in automotive networks on three main aspects:

- Encryption of the communications
- Anomaly detection
- Integrity of the embedded software

Research on such topics is really intense today, as evidenced by the strong implication of manufacturers and academics into several large-scale projects whose results enable the current implementations of first security modules. However, there is still many work to do, especially as experiences from traditional computing remind us that such issues may never be completely solved.

[5]http://www.xtratum.org/

ACKNOWLEDGEMENTS

This study is partially supported by ANRT Convention CIFRE n°2012/0189. The authors are grateful to the anonymous reviewers for their help in improving this paper.

REFERENCES

[1] R. N. Charette, "This car runs on code," *IEEE Spectr.*, vol. 46, no. 3, p. 3, 2009.

[2] L. D'Orazio, F. Visintainer, and M. Darin, "Sensor networks on the car: State of the art and future challenges," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, 2011, pp. 1–6.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. 20th USENIX Security*, San Francisco, CA, 2011.

[4] R. Brooks, S. Sander, J. Deng, and J. Taiber, "Automobile security concerns," *IEEE Veh. Technol. Mag.*, vol. 4, no. 2, pp. 52–64, 2009.

[5] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, 2007.

[6] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Laboratories, Tech. Rep. SAND98-8667, 1998.

[7] V. Verendel, D. K. Nilsson, U. E. Larson, and E. Jonsson, "An approach to using honeypots in in-vehicle networks," in *68th Vehicular Technology Conf.* Calgary, Canada: IEEE, 2008, pp. 1–5.

[8] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *2nd Embedded Security in Cars Workshop (ESCAR 2004)*, Bochum, Germany, 2004, pp. 11–12.

[9] T. Hoppe, S. Kiltz, and J. Dittmann, "Automotive it-security as a challenge: Basic attacks from the black box perspective on the example of privacy threats," *Computer Safety, Reliability, and Security*, pp. 145–158, 2009.

[10] T. Hoppe and J. Dittman, "Sniffing/replay attacks on can buses: A simulated attack on the electric window lift classified using an adapted cert taxonomy," in *Proc. 2nd Workshop on Embedded Systems Security (WESS)*, Salzburg, Austria, 2007.

[11] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, and H. Shacham, "Experimental security analysis of a modern automobile," in *2010 IEEE Symp. Security and Privacy*, Oakland, CA, 2010, pp. 447–462.

[12] D. K. Nilsson and U. E. Larson, "Simulated attacks on can buses: vehicle virus," in *Proc. 5th IASTED Int. Conf. on Communication Systems and Networks*, Langkawi, Malaysia, 2008, pp. 66–72.

[13] R. Moalla, H. Labiod, B. Lonc, and N. Simoni, "Risk analysis study of its communication architecture," in *3rd Int. Conf. Network of the Future*, Tunis, Tunisia, 2012, pp. 1–5.

[14] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *Proc. USENIX Security Symposium*, Washington, DC, 2010, pp. 323–338.

[15] N. Courtois, G. Bard, and D. Wagner, "Algebraic and slide attacks on keeloq," in *Fast Software Encryption*. Lausanne, Switzerland: Springer, 2008, pp. 97–115.

[16] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. Shalmani, "On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme," *Advances in Cryptology*, pp. 203–220, 2008.

[17] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," *IACR ePrint Report*, vol. 2010/332, 2010.

[18] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Intelligent Vehicles Symposium (IV)*. Baden Baden: IEEE, 2011, pp. 528–533.

[19] K. Scarfone and J. Padgette, "Guide to bluetooth security," *NIST Special Publication*, vol. 800, p. 121, 2008.

[20] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung *et al.*, "Secure vehicular communication systems: implementation, performance, and research challenges," *Communications Magazine*, vol. 46, no. 11, pp. 110–118, 2008.

[21] "About PRESERVE," http://www.preserve-project.eu/about, 2011, [Online; accessed February-2013].

[22] O. Henniger, A. Ruddle, H. Seudié, B. Weyl, M. Wolf, and T. Wollinger, "Securing vehicular on-board it systems: The evita project," in *25th VDI/VW Automotive Security Conf.*, Ingolstadt, Germany, 2009.

[23] A. Groll, J. Holle, C. Ruland, M. Wolf, T. Wollinger, and F. Zweers, "Oversee a secure and open communication and runtime platform for innovative automotive applications," in *7th Embedded Security in Cars Conf. (ESCAR)*, Düsseldorf, Germany, 2009.

[24] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *9th Embedded Security in Cars Conf.*, Dresden, Germany, 2011.

[25] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede, "Libra-can: a lightweight broadcast authentication protocol for controller area networks," in *Proc. 11th Int. Conf. Cryptology and Network Security, CANS*, Darmstadt, Germany, 2012.

[26] O. Hartkopp, C. Reuber, and R. Schilling, "Macan - message authenticated can," in *10th Int. Conf. on Embedded Security in Cars (ESCAR 2012)*, 2012.

[27] M. Wolf and T. Gendrullis, "Design, implementation, and evaluation of a vehicular hardware security module," *Information Security and Cryptology-ICISC 2011*, pp. 302–318, 2012.

[28] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "Car2x communication: securing the last meter-a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography," in *Vehicular Technology Conf. (VTC Fall)*. San Francisco, CA: IEEE, 2011, pp. 1–5.

[29] I. Broster and A. Burns, "An analysable bus-guardian for event-triggered communication," in *Real-Time Systems Symposium*. IEEE, 2003, pp. 410–419.

[30] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, and K. Oishi, "A method of preventing unauthorized data transmission in controller area network," in *Vehicular Technology Conf. (VTC Spring)*. Yokohama, Japan: IEEE, 2012, pp. 1–5.

[31] H. Schweppe and Y. Roudier, "Security and privacy for in-vehicle networks," in *Vehicular Communications, Sensing, and Computing (VCSC)*. Seoul, Korea: IEEE, 2012, pp. 12–17.

[32] M. Muter, A. Groll, and F. C. Freiling, "A structured

approach to anomaly detection for in-vehicle networks," in *6th Int. Conf. Information Assurance and Security (IAS)*. Atlanta, GA: IEEE, 2010, pp. 92–98.

[33] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Intelligent Vehicles Symposium (IV)*. Baden Baden, Germany: IEEE, 2011, pp. 1110–1115.

[34] W. A. Arbaugh, D. J. Farber, and J. M. Smith, "A secure and reliable bootstrap architecture," in *Proc. Symp. Security and Privacy*. Oakland, CA: IEEE, 1997, pp. 65–71.

[35] "TPM main specification," http://www.trustedcomputinggroup.org/resources/tpm_main_specification, 2011, [Online; accessed February-2013].