REAACT: TESTBED FOR VIRTUAL VEHICLES

Jan-Alexandru VĂDUVA[1], Dragoș-Florin CURTEANU[2], Răzvan RUGHINIȘ[3]

There is an increased need for cybersecurity professionals with specialized knowledge in line with the regulatory requirements of R156 and R155 issued by the United Nations Economic Commission for Europe (UNECE), given the growing cyber threats in cyber-physical systems, especially in the automotive industry. The development of testbeds that can simulate a vehicle's control system, sensors, and actuators is essential for addressing this requirement. This study introduces the first REference Automotive ArChitecture Testbed (REAACT) prototype, which is intended to carry out this essential function.

The REAACT testbed explores the reliability of software-based steering and braking systems inside the vehicle. By doing so, we want to gather important facts about the impact of these advanced software control methods have on the dependability and safety of cars. Using an open-source 3D automobile racing simulator, we construct a network of connected components that each perform control strategies like traction control and braking aid. This simulator contains an integrated virtual machine that enables task tracking and updating while additionally allowing inputs from the keyboard. Furthermore, we adjust control parameters using data analysis to enhance the performance and safety of the systems.

Keywords: testbed, software control algorithms, vehicle safety, simulation

1. Introduction

Automobile manufacturers can no longer consider the installed systems to be isolated due to the enormous progress being made in expanding the connection of automotive systems. The concerns about cybersecurity for automobile systems have, in fact, grown significantly over the last few decades. Existing investigations [1] and cyberattacks [2] serve as examples of the visible dangers that both physical and remote attacks on automobiles represent. Along with other security issues for vehicles, recent research studies [3] have also emphasized the significance of guaranteeing the security. The daily operations of a car could be negatively impacted by a malicious attack, which could have severe consequences. Testing the in-vehicle network is therefore essential to identify any security vulnerabilities.

[1] Lecturer, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: jan.vaduva@upb.ro

[2] Student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: dragos.curteanu@stud.electro.upb.ro

[3] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: razvan.rughinis@upb.ro

The electrical vehicles have made a significant impact on the automotive industry in recent years, thanks to their great potential to cut carbon dioxide emissions and decrease dependence on fossil fuels. Emerging propulsion innovations, like in-wheel motors, which have attracted a lot of attention for their ability to boost vehicle efficiency, performance, and mobility, are at the core of this revolution. To control and optimize the complex interactions of power distribution, vehicle dynamics and regenerative braking the switch to electric engines and in-wheel motors, however, requires a substantial increase in software systems and lines of code. We estimate that the new electrical EQS models have already surpassed the 200 million lines of code estimate made by the business research firm Frost & Sullivan since the 2009 Mercedes-Benz S-Class, which had according to [4], between 70 and 100 micro-controller-based electronic control units (ECUs) connected across the vehicle. The software complexity for those vehicles equipped with the latest innovations, including in-wheel motors, is only imaginable.

The REAACT testbed's primary goal is to carefully analyze and evaluate the effects of integrating complex software control algorithms into the areas of the vehicle that are most crucial for safety: the steering and braking systems. We present a case study of practical interest for the automotive industry where we could explore formal verification, design-space exploration, and simulation. Our primary objective is to get important insights regarding the impact of these advanced software control techniques on vehicle reliability and safety through an extensive research effort.

The proposed strategy involves employing a virtual simulator to simulate both the car and its associated settings as well as modeling both the electronic control units (ECUs) for braking and acceleration independently with two hardware components. Both are linked by a third hardware component which is also responsible for the communication with the simulation. The suggested approach entails using a virtual simulator to recreate the vehicle and all its settings as well as modeling the electronic control units (ECUs) for braking and acceleration separately. A third piece of hardware, which connects both, is also in charge of communicating with the simulation. The advantage of the simulation is the speed it gave in this exploratory process. Even though there are numerous publicly available virtual simulators, including open-source alternatives, which may result in less accurate conclusions when compared to real-world approaches they can be configured to mimic as much as possible the real-world scenarios.

The paper is set up as follows. An overview of the problem and automotive systems is provided in Section 2 of this article. In Section 3, a survey of relevant literature on automobile cybersecurity testbeds is discussed. In Section 4, we outline the methodology and procedure of our research. Section 5 presents the research results. Finally, conclusions are given in Section 6.

2. Background

Because there are more ECUs and sub-networks linking them, automotive systems have grown more sophisticated in the past few years [11, 37, 38]. The controller area network (CAN), the FlexRay, the local interconnect network (LIN), and the media-oriented systems transport (MOST) sub-networks are all parts of the automotive network. Regarding the application in various car models, each sub-network has specific advantages in terms of bus speed, trigger type, and benefits. This paper does not discuss these technological specifics and benefits, but they are available in articles like [11, 39].

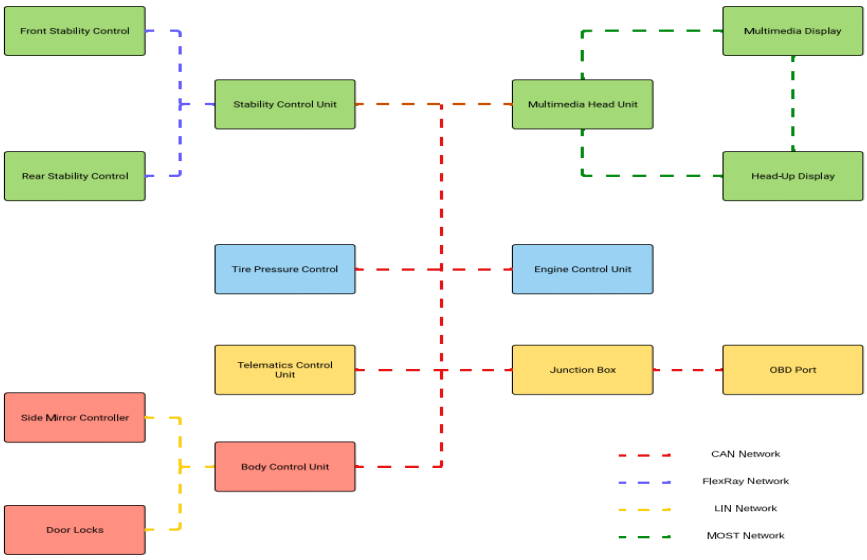

Fig. 1. The automotive sub-networks

Although this is not generally applicable, each of the above-mentioned sub-networks has their very specific role. The CAN sub-network for example is the backbone in almost all vehicles and through the on-board diagnostic (OBD) port it provides a standardized way of testing. The LIN sub-network is usually available for low-speed vehicle body interactions. For the multimedia automotive applications usually the MOST sub-network is used, especially lately since it received support inside the Linux kernel. While the safety-critical applications such as stability control are managed by FlexRay. In a nutshell, each offers specific benefits and drawbacks that automobile manufacturers can take advantage of to lower total production costs while delivering the highest network performance.

Each sub-network complies with approved industry standards, on top of those the automotive companies can extend and implement application layers which are typically not accessible to the broader public. It is as described by the AUTOSAR working principles of “Cooperate on standards, compete on

implementation", which makes the job of the researchers and engineers extremely difficult and involves a lot of reverse-engineering to extract the relevant information.

3. Related work

This section introduces and discusses prior research on vehicle security testbeds. They are available as a full software solution [5] without the use of a vehicle, there are also full hardware setups both involving the use of lab configurations [8, 9] or real-world sets [6, 7]. Most of them were restricted to the particular security testing methodology that was investigated. There isn't a platform for testing that is both inexpensive and very flexible for researchers. High-precision automotive simulation testbeds are currently available; however, they are costly, limited in functionality, and hard to duplicate.

The previous research presents a summary of the security challenges with automotive networks [10, 11, 12, 14, 15, 16, 17, 18], of the threats and attacks possible [14, 31] and provides solutions under the form of firewalls [11], intrusion detection systems [26, 27, 28, 29, 30], honeypots [26, 32], encrypted and secure communication [33, 34, 35, 36, 11, 37] together with architectural decisions [10, 19, 20, 21, 22, 23, 24, 25] available to mitigate the risks to safety and security.

By contract REAACT is an open-source, cost-effective, versatile, and portable automotive reference architecture which serves as a platform able to analyze modern automotive technology and safety innovations, offering accessibility, adaptability, and reproducibility for research and development purposes.

4. Testbed

The method of choice involves modeling both the car and the associated configuration in a virtual simulator. This enables precise control of every aspect of the car and quick access to its data, making system design and testing much simpler. There are many virtual simulators easily available right now, some of them are open-source solutions. The disadvantage, though, is that the results would not be as precise as with the real-world approaches such as a complete vehicle or vehicular subsystems. However, this strategy provides a substantially less expensive option.

The electronic control units (ECUs) for brakes and acceleration were modelled independently from additional hardware parts for the most realistic integration of systems using the simulator. A third piece of hardware served as the interface between these ECUs and the simulator.

4.1. Hardware

There are three components to the hardware structure. The gateway or we can also call it the central compute module is responsible to the simulator connection and communication. It handles all that data, delivers it to the other vehicle ECUs and returns the adjusted results from them. All this is applied to the simulator vehicle. The other two components are the brake and throttle ECUs which are responsible for the algorithm application and justification for controlling independently the break and throttle outputs.

The gateway ECU is based on a Raspberry Pi 3 Model B+ and a hardware CAN extension which supports two CAN connections at once enabling asynchronous communication between itself and the other ECUs. For the rest of hardware components, we chose to use an MKR CAN and an MKR Zero combination, where the MKR CAN is the extension enabling one CAN communication interface on top of the Arduino MKR Zero.

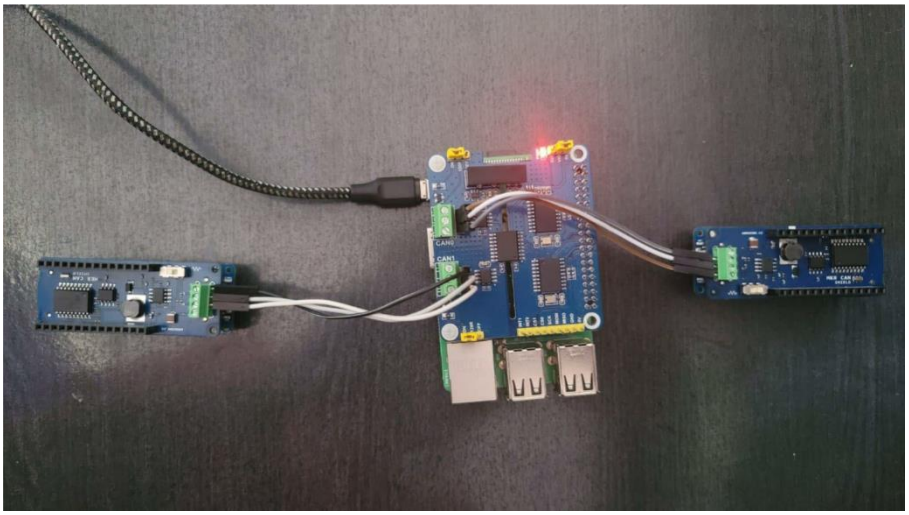

Fig. 2. The hardware components

The above-described electronic devices have been set up to preserve the vehicle's smooth movement. As part of this setup, we can include the anti-braking system, or ABS, ensures consistent and efficient deceleration without locking the car's wheels, allowing drivers to maintain control of their path even during severe braking. We enabled traction control which ensures uniform and efficient deceleration without the tires losing traction and spinning in place, preventing you from losing your grip on the car in the event of rapid acceleration. And finally, stability control, which prevents the car's wheels from slipping during quick turns at high speeds, ensures that the intended path will remain unchanged.

4.2. Software

The Open Racing Car Simulator (TORCS), an open-source 3D simulator that is also accessible on Linux, was selected because of its comprehensive documentation. It supports the development of a customizable driver, and because it is implemented in C++, it proved straightforward to get the needed information out of the simulator and transfer it to the gateway unit via the driver's code. It also provides a debug possibility, which is incredibly helpful. Because of the bespoke driver, it is very adaptable, making the integration and testing of the set-up solutions accessible.

First, we had to do introduce the option of individually controlling the simulator for starting the experimentation with individual motor control for brake, steer, and acceleration:

```
+   bool individualBrakes;
+   tdbleindividualBrakeCmd[4];
.....
+#define _individualBrakes ctrl.individualBrakes
+#define _individualBrakeCmd ctrl.individualBrakeCmd
.....
+   if(!car->ctrl->individualBrakes)
+   {
.....
+   else
+   {
+      tdble *ctrl = car->ctrl->individualBrakeCmd;
+
+      car->wheel[FRNT_LFT].brake.pressure = ctrl[0]*brkSyst->coeff;
+      car->wheel[FRNT_RGT].brake.pressure = ctrl[1]*brkSyst->coeff;
+      car->wheel[REAR_LFT].brake.pressure = ctrl[2]*brkSyst->coeff;
+      car->wheel[REAR_RGT].brake.pressure = ctrl[3]*brkSyst->coeff;
+   }
```

Here we can see how the change for the individual breaking looks like and this was essential for enabling the experimentation and safety evaluation of these software-based control algorithms.

With the simulator functionality appended we then made sure that we can extract the relevant car parameters with the help of a driver implementation. Next step was enabling the interaction with the simulator which was done through keyboard interaction and an easy keylogger implementation. The use had the following actions available:

With all this done we continued with the communication support between the gateway and the other ECUs. We initially check if we have active systems to interact with and then we proceed with data transmission. The data needed for braking adjustments is prepared and provided as CAN signals to the ECU, indicating which systems were active, if the user engages either the ABS or Stability Control systems. Like this, data for acceleration adjustments is prepared and sent to the appropriate ECU if the traction control system is engaged. Based on the user-activated systems, the ECUs for braking and acceleration calculate the necessary

modifications and prepare them as CAN messages to send back to the gateway. The gateway then sends these computed adjustments to TORCS for implementation after receiving them. The final step consists of the software implementation of the ECUs for accelerating and braking, specifying calculating modes for modification depending on user-activated systems and outlining specific adjustments for each system.

If the researcher activates ABS, modifications to the brakes involve recalculating brake pressure settings for each wheel based on variances in wheel and vehicle speed. If Stability Control is engaged, a brake pressure correction based on vehicle rotation speed is computed and, depending on the vehicle's rotation speed, this correction is applied to the rear left or right wheel. When the user engages Traction Control, the ECU must calculate a correction for acceleration pressure depending on the speed of the vehicle's wheels at the back and the vehicle itself. An acceleration adjustment is also computed if Stability Control is engaged.

5. Research and Teaching Opportunities

With the ability to activate numerous systems simultaneously for testing, implemented systems can be readily evaluated through the simulator. Certain behaviors can be seen even without any systems being active. By turning on the ABS system, which modifies the brake pressure on the wheels during braking to prevent them from locking, the problem can be resolved. Like this, when moving quickly and continuously from a stop, the wheels lose traction, the car spins out of control, and the driver loses control. By turning on the traction control system, which continuously modifies acceleration to minimize wheel slippage and ensures user control during acceleration, this issue can be lessened. Sharp maneuvers made while accelerating at a high speed may cause the wheels to begin to skid, which could cause the car to spin out of control and the driver to lose control. By turning on the stability control system, this problem can be solved. The results gained meet the required criteria, and the technologies put in place give the operator better control of the car and a steadier trajectory.

Researchers can gain important new understanding of behaviors like wheel lockup during braking and wheel slippage during acceleration thanks to the simulator's extensive platform for in-depth investigation and testing with numerous automobile systems. Beyond performance assessment, this capacity creates opportunities for research into the security and safety features of electronic implementations, particularly those aiming for the most stringent compliance.

Using the simulator offers users a rare educational opportunity to investigate the integration of safety and security measures within electronic systems involved in the electrical powertrain control in addition to hands-on involvement to enhance driving dynamics by activating systems like ABS, Traction Control, and Stability

Control. This emphasizes the significance of strong security features in modern vehicle electronics and is particularly significant in the context of obtaining Automotive Safety Integrity Level D (ASIL D) compliance.

6. Future Work and Conclusion

By integrating ABS, Traction Control, and Stability Control using Raspberry Pi and Arduino-based ECUs in the TORCS simulator, we can design, implement, and test automated systems that improve vehicle stability and provide better driver control. Future objectives call for the addition of safety systems like Adaptive Cruise Control and Brake Assist for further enhancements to the project.

In the future, there is a compelling opportunity to explore the convergence of Artificial Intelligence (AI) and determinism in achieving an ASIL D solution for comprehensive electronic control of braking, steering, and acceleration in electric vehicles. This pursuit holds significant potential for both research and teaching purposes. It allows educators and students to engage with the latest advancements in automotive technology and safety. This hands-on approach not only prepares future engineers for the automotive industry but also contributes to the broader discourse on the ethical and technological dimensions of autonomous systems.

Acknowledgement

"This work was supported by the grant POCU/993/6/13 -153178, co-financed by the European Social Fund within the Sectorial Operational Program Human Capital 2014 – 2020".

R E F E R E N C E S

[1]. *Zhang, H., Huang, K., Wang, J. and Liu, Z., 2021, September.* CAN-FT: A Fuzz Testing Method for Automotive Controller Area Network Bus. In 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI) (pp. 225-231). IEEE.

[2]. *Popa, L., Berdich, A. and Groza, B., 2022.* CarTwin—Development of a Digital Twin for a Real-World In-Vehicle CAN Network. Applied Sciences, 13(1), p.445.

[3]. *Limbasiya, T., Teng, K.Z., Chattopadhyay, S. and Zhou, J., 2022.* A systematic survey of attack detection and prevention in connected and autonomous vehicles. Vehicular Communications, p.100515.

[4]. *Charette, R.N., 2009.* This car runs on code. IEEE spectrum, 46(3), p.3.

[5]. *J. Munera, J. M. de Fuentes, A. I. González-Tablas,* Towards a comparable evaluation for VANET protocols: NS-2 experiments builder assistant and extensible test bed, escar Europe 2011, 2011.

[6]. *K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage,* Experimental Security Analysis of a Modern Automobile, IEEE Symposium on Security and Privacy, 2010.

[7]. *K. Fischer,* High Assurance Cyber Military Systems (HACMS), escar USA2013, 2013

[8]. *C. Miller, C. Valasek,* Car Hacking: For Poories, SyScan2014, 2014.

[9]. *T. Toyama, T. Yoshida, H. Oguma, and T. Matsumoto*, Pasta: portable automotive security testbed with adaptability, London, blackhat Europe, 2018.
[10]. *D. K. Nilsson and U. E. Larson*. Simulated Attacks on CAN Buses: Vehicle Virus. In: Proceedings of the 5th IASTED International Conference on Communication Systems and Networks. AsiaCSN '08. Anaheim, CA, USA. Palma de Mallorca, Spain: ACTA Press, 2008, pp. 66–72.
[11]. *M. Wolf, A. Weimerskirch and C. Paar*. Security in Automotive Bus Systems. In: Workshop on Embedded IT-Security in Cars. Bochum, Germany, Nov. 2004.
[12]. *K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway et al*. Experimental Security Analysis of a Modern Automobile. In: 2010 IEEE Symposium on Security and Privacy (SP). IEEE, 2010, pp. 447–462.
[13]. *C. Sandberg*, HoliSec Automotive Security and Privacy Holistic Approach to Improve Data Security, March 2003.
[14]. *T. Hoppe and J. Dittmann*. 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.
[15]. *A. Lang, J. Dittmann, S. Kiltz and T. Hoppe*. Future Perspectives: The Car and Its IP-Address — A Potential Safety and Security Risk Assessment. In: Proceedings of the 26th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '07). SAFECOMP '07. Nuremberg, Germany, Sept. 2007, pp. 40–53.
[16]. *T. Hoppe, S. Kiltz and J. Dittmann*. Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures. In: Computer Safety, Reliability, and Security. Ed. by M. D. Harrison and M.-A. Sujan. Lecture Notes in Computer Science 5219. Springer Berlin Heidelberg, Jan. 2008, pp. 235–248.
[17]. *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. In: Computer Safety, Reliability, and Security. Ed. by B. Buth, G. Rabe and T. Seyfarth. Lecture Notes in Computer Science 5775. Springer Berlin Heidelberg, Jan. 2009, pp. 145–158.
[18]. *D. K. Nilsson, U. E. Larson, F. Picasso and E. Jonsson*. A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay. In: Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS'08). Ed. by E. Corchado, R. Zunino, P. Gastaldo and Á. Herrero. Vol. 53. Advances in Intelligent and Soft Computing. Springer Berlin / Heidelberg, 2009, pp. 84–91.
[19]. *M. L. Chávez, C. H. Rosete and F. R. Henríguez*. Achieving Confidentiality Security Service for CAN. In: Proceedings of the 15th International Conference on Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Feb. 2005, pp. 166–170.
[20]. *D. K. Nilsson, U. E. Larson and E. Jonsson*. Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes. In: Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th. 2008, pp. 1–5.
[21]. *A. Groll and C. Ruland*. Secure and Authentic Communication on Existing In-Vehicle Networks. In: 2009 IEEE Intelligent Vehicles Symposium. 2009, pp. 1093–1097.
[22]. *H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka and H. Imai*. New Attestation-Based Security Architecture for In-Vehicle Communication. In: IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. New Orleans, Louisiana, 2008, pp. 1–6.
[23]. *C. Szilagyi and P. Koopman*. A Flexible Approach to Embedded Network Multicast Authentication. In: 2nd Workshop on Embedded Systems Security (WESS). 2008.
[24]. *S. Schulze, M. Pukall, G. Saake, T. Hoppe and J. Dittmann*. On the Need of Data Management in Automotive Systems. In: 13. Fachtagung des GI-Fachbereichs "Datenbanken und

Informationssysteme" (DBIS). Vol. 144. Gesellschaft für Informatik (GI). Münster, Germany, Mar. 2009.

[25]. *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: 2011 IEEE Vehicular Technology Conference (VTC Fall). 2011, pp. 1–5.

[26]. *U. E. Larson and D. K. Nilsson*. Securing Vehicles against Cyber Attacks. In: CSIIRW '08: Proceedings of the 4th annual workshop on Cyber security and information intelligence research. CSIIRW '08. Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead. New York, NY, USA: ACM, 2008, 30:1–30:3.

[27]. *T. Hoppe, S. Kiltz and J. Dittmann*. Adaptive Dynamic Reaction to Automotive IT Security Incidents Using Multimedia Car Environment. In: Proceedings of the 4th International Conference on Information Assurance and Security (ISIAS '08). Sept. 2008, pp. 295–298.

[28]. *T. Hoppe, S. Kiltz and J. Dittmann*. Applying Intrusion Detection to Automotive IT — Early Insights and Remaining Challenges. In: Journal of Information Assurance and Security 4.3 (2009), pp. 226–235.

[29]. *M. Müter, A. Groll and F. C. Freiling*. A Structured Approach to Anomaly Detection for In-Vehicle Networks. In: 2010 Sixth International Conference on Information Assurance and Security (IAS). Atlanta, GA, Aug. 2010, pp. 92–98.

[30]. *M. Müter and N. Asaj*. Entropy-Based Anomaly Detection for In-Vehicle Networks. In: 2011 IEEE Intelligent Vehicles Symposium (IV). Baden-Baden, Germany, June 2011, pp. 1110–1115.

[31]. *R. Brooks, S. Sander, J. Deng and J. Taiber*. Automobile Security Concerns. In: Vehicular Technology Magazine, IEEE 4.2 (June 2009), pp. 52–64. ISSN: 1556-6072.

[32]. *V. Verendel, D.K. Nilsson, U.E. Larson, and E. Jonsson*. An approach to using honeypots in in-vehicle networks. In Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, pages 1–5, 2008.

[33]. *B. Groza and S. Murvay*. Secure broadcast with one-time signatures in controller area networks. In Availability, Reliability and Security (ARES), 2011 Sixth International Conference on, pages 371–376, aug. 2011.

[34]. *G. Lee, H. Oguma, A. Yoshioka, R. Shigetomi, A. Otsuka, and H. Imai*. Formally verifiable features in embedded vehicular security systems. In Vehicular Networking Conference (VNC), 2009 IEEE, pages 1–7, 2009.

[35]. *K. Lemke, C. Paar, and M. Wolf, editors*. Embedded Security in Cars: Securing Current and Future Automotive IT Applications. Springer Publishing Company, Incorporated, 1st edition, 2006.

[36]. *H. Schweppe and Y. Roudier*. Security and privacy for in-vehicle networks. In Vehicular Communications, Sensing, and Computing (VCSC), 2012 IEEE 1st International Workshop on, pages 12–17, june 2012.

[37]. *M. Wolf, A. Weimerskirch, and T. Wollinger*. State of the art: Embedding security in vehicles. EURASIP Journal of Embedded Systems, 2007.

[38]. *G. Leen and D. Heffernan*. Expanding automotive electronic systems. Computer, 35(1):88–93, 2002.

[39]. *D. Paret*. Multiplexed Networks for Embedded Systems. John Wiley and Sons, Ltd, 2007.