Copy Protection for Automotive Electronic Control Units using Authenticity Heartbeat Signals

Armin Wasicek
Institute for Computer Engineering
Vienna University of Technology, Austria
Email: armin.wasicek@tuwien.ac.at

***Abstract*—Protection of intellectual property rights is a vital aspect for the future automotive supplier market, in particular for the aftersales market for ECUs. Computer security can deliver the required protection mechanisms and sustain the according business models. We propose an approach to facilitate the rigorous checking of components for originality in a vehicle. In our system model, a security controller receives special messages (i.e., the authenticity heartbeat signal) from relevant ECUs and it performs subsequent authentication and plausibility checks. As a result, the security controller can tell, if the current setup of components in the vehicle is original. We evaluate our authentication architecture for the Battery Management System (BMS) of a hybrid car. Here, the security controller detects reliably, if the BMS is an original component, and whether an attacker has modified the operational limits of the battery. In this paper, we reason that an effective copy protection scheme needs to fuse relevant information from different sources. Therefore, various security techniques have to be combined in a sound architectural approach. The distinctive feature of our architecture is that it takes into account application–specific knowledge of the real–time entities under control.**

I. Introduction

Software offers application designers a tremendous flexibility to design, implement, and maintain their products. This flexibility motivates the automotive industry to shift towards software–centric vehicles [13]. The benefits of software entail certain drawbacks. Software has no inherent mechanisms to protect itself from unauthorized modifications or illegal copies. Illegal copies are a recognized problem in the industry. The Business Software Alliance (BSA) amounts the global value of unlicensed software in 2010 to $58.8 billion [4]).

Software piracy is a particular threat for innovative hi–tech companies. Their competitive advantage on the global market is mainly based on their cutting edge know–how. This know–how is often delivered to the customers in form of software. Companies and organizations disrespecting Intellectual Property (IP) protection laws and distributing copied software can avoid investments on R&D and threaten the markets by selling counterfeit products for a fraction of the usual price. The prevention of using counterfeit components and software is not only relevant for economical reasons, but also for safety, because these imitations might not conform to international safety standards.

Combating counterfeiting is a tough challenge, because counterfeiting is a multi-dimensional problem involving technological, social, legal, political, and commercial issues. On the technological side, protection mechanisms for software usually require at least minimal support from the hardware to establish a so–called *root–of–trust* [25]. This has to be supported by the platforms used. Moreover, solutions have not only to be technically secure, but also they have to be organizationally feasible and aligned to the life cycle of a product.

Current approaches to deliver copy protection fail, because they are addressing only a part of the problem or because they are technically not feasible in the context of a vehicle. Particular concerns that hamper the implementation of standard ICT security solutions in an automotive environment encompass stringent resource constraints (limited processing power and small memory sizes), real–time requirements on communication and computation of algorithms, deployment in an untrusted environment (the owner could be the attacker), and cost reasons [10].

In this paper we focus on preventing the use of illegal copies of automotive Electronic Control Units (ECUs) in order to retain the competitive advantage of hi–tech industries and to sustain safety requirements. Our key contribution is the definition of an authentication protocol called *Authenticity Heartbeat Signal* to continuously monitor the composition of ECUs in a vehicle. In particular, our contribution focuses on:

- Motivate ECU copy protection through a market analysis
- Develop an authentication architecture for ECUs including the specification of the Authenticity Heartbeat Signal
- Evaluate the concepts in a hybrid car setup

Our solutions are capable of enforcing societal rules, legal regulations, and business models based on copy rights by technological means (i.e., security mechanisms and protocols).

In Section II we analyze the relevant organizational, economical, and technical aspects of the automotive environment. Next, we give an account of existing approaches for copy protection in Section III. Our solution proposes to use a combination of authentication and plausibility checking techniques in Section IV. Section V evaluates the concepts in a hybrid car setup. In Section VI we reference related work. The paper concludes in Section VII.

II. Automotive Environment

The automotive environment comprises the vehicle and all entities contributing to produce, operate, and maintain the vehicle. Figure 1 depicts a simplified view on the automotive

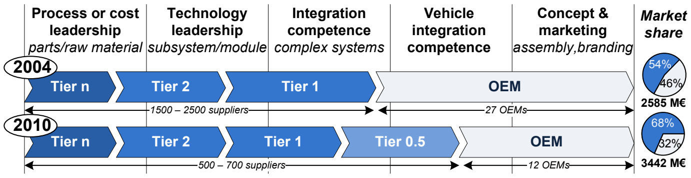

Fig. 1. Automotive value and supply chain

value and supply chains in 2002 and an outlook to 2015. In–depth analysis of the automotive environment is subject of many publications [26] [7] [12].

A. Automotive Value Chain

Currently, the automotive industry is undergoing substantial consolidation process. Starting with 36 car manufacturers in 1970, each having a different specialization, today there are 12 globally acting Original Equipment Manufacturers (OEMs), each addressing the full range of vehicles [26]. At the same time, the supply chain is consolidating. The traditional first–tier and second–tier suppliers are undergoing substantial merging process and a few global mega–suppliers, sometimes referred to as ”Tier 0.5 suppliers”, will remain. This new supplier class will deliver complete vehicle subsystems (e.g., a power train). Moreover, it will have vehicle integration competences which were up to now allocated to OEMs. Therefore, the suppliers' market share will increase to about 68% in 2015.

Alongside with these mergers, profit margins from car sales are decreasing. The after–sales market is becoming an increasing important source of revenue for many OEMs. For example, in Germany the after–sales business generates more than half of an OEM's profits while accounting for only 23% of an OEM's revenues [6].

The after–sales market is not restricted to OEMs and mega–suppliers, but competitors will be found across remaining first–tier and second–tier suppliers. Aftermarket products will be traded internationally and the internet will emerge as a retail channel, enabling the sale of low-price parts and direct customer relations. Competition will be based predominately on price. Software will play a major role in the development of business cases and as a product for the after–sales market. For example, Ford (an OEM) anticipates a software update cycle for vehicles of 6 months [13].

B. Increasing Electronics

A further trend in the automotive industry is the increase of electronics in vehicles from 19 % in 2004 to 40 % in 2015. This trend has several reasons: First, *electronic components replace mechanical parts*. For instance in a drive–by–wire system, an ECU wired with sensors and actuators replaces the mechanical link (a Bowden cable) between pedal and throttle. The benefits are improved ergonomics and increased reliability. Second, *new applications* are realized to add new features. According to Audi (an OEM), electronics and software will enable 90 % of all future innovations [20]. Examples for new systems are driver information and assistance systems, intelligent lighting systems, night vision systems, adaptive cruise control, and comfort systems [26]. Third, vehicle electronics sustains *new safety functions* like Anti–lock Braking System (ABS) and Electronic Stability Program (ESP). Fourth, electronics enable *legal obligations* like emission control.

C. Vehicle networks

A modern car contains approximately to 70 ECUs which are connected on various in–vehicle networks (sometimes called VANets). They realize automotive subsystems like power train, fuel–delivery, or electrical control. The different in–vehicle networks employ a multitude of communication protocols (e.g., CAN, LIN, FlexRay) with different properties like transmission speed, bandwidth, real–time capabilities, or fault tolerance. Traditional automotive architectures employ a different network for each subsystems. Current approaches aim to integrate different subsystems to increase efficiency [15] [9].

D. Chip Tuning

Chip tuning attacks aim at change the behavior of a vehicle's control algorithms by modifying the vehicle's software. Control algorithms are mostly implemented as software in a vehicle's ECU. The critical parts of a control application are the parameters of the control algorithms. Usually, these parameters are stored in a table within an ECU's flash memory. Finding the location of this table in the memory of the ECU facilitates the modifications of the engine's control parameters and hence altering the ECU's functionality. For example, a popular tuning attack is to increase the engine's horse power by manipulating the engine's ECU.

A chip tuning attack executes as follows: The tuner reads out the ECU's flash memory to produce a raw binary dump of the control application's binary image. This can be done, for instance, using the On–Board Diagnostics (OBD) interface [21] of an ECU. Next, the image is disassembled to reveal the structure of the code. The next step is to locate and modify the desired parameters. This is often done in a try–and–error fashion. The modified image is then reprogrammed in the original ECU's flash memory. Various companies offer chip tuning as a service. These companies assist vehicle owners or they work on behalf of the owners to carry out chip tuning.

Chip tuning raises concerns about the warranty for a chip tuned vehicle and its impact on safety and certification. Moreover, the used binary dump of the ECU's application represents an illegal copy, because its usage is usually not authorized by the OEM. Security techniques like for example Secure Boot [11] [29] can effectively counteract chip tuning attacks. A restriction is that if the parameters are modified after the boot process is completed (e.g., by overwriting the parameters in the volatile memory), Secure Boot schemes cannot help. However, chip tuning is not yet on the radar of big automotive companies, because chip tuning is tedious and the number of practitioners is low.

E. Powerboxing

Powerboxing [19] directly modifies the output signals of an ECU by inserting a hardware module in the vehicle. The rationale behind this idea is similar to chip tuning: an attacker wants to tap the engine's maximum potential. The inserted module either replaces the original ECU on the communication system or it is placed as a man–in–the–middle between the original ECU and its connected actuators. The installation in the vehicle is fairly simple as demonstrated by several supplier guidelines and demonstration videos on the internet. It requires plugging the new unit to the power supply and to reconnect the network cables from the original ECU to the powerboxing module.

F. Counterfeit ECUs

A counterfeit ECU is a technically correct ECU which has been produced by a third party without authorization. Powerboxing shows that it is feasible for a third party to integrate functioning ECUs in a vehicle's computer system. Obscuring the technical details of the implementation does not help. Poxerboxing is often done by a third party that does not necessarily have all the required design information of the target vehicle at hand.

The assembly of a counterfeit ECU involves a reverse engineering of the ECU's hardware. The forged hardware can then be programmed similarly to a chip tuning attack by simply reading out a binary image from an original ECU and flashing this image to the counterfeit one. However, the quality of the counterfeit ECU might not meet international production standards or electromagnetic compatibility, because the counterfeiter is likely to operate hidden and to deny liability in case of failure.

G. Summary

In this paper, we highlight computer security methods and techniques to detect and prevent the unauthorized insertion of a chip tuned, powerboxed, or counterfeit ECU in a vehicle. We believe that our work is important for following reasons:

- Safety and certification issues are unclear for modified vehicles. Therefore, the OEM should be protected by a kind of *electronic warranty seal* enabled by the security mechanisms.
- Many low–tech spare parts (e.g., parts of the chassis) are already copied by third party manufacturers. The appearance of hi–tech parts like ECUs on the grey market is just a matter of time. The assembly of counterfeit ECUs represents a clear violation to an OEM's or a supplier's *IP* rights.
- The fact that the after–sales market yields high returns makes it attractive for new entrants. In order to *keep the market balanced*, the entry barrier with respect to know–how must be kept even for all competitors. By copying existing products, a new competitor could lower this barrier and ruin the market for companies having invested on their know–how.

III. ECU Copy Protection Schemes

In this section we survey the current academic and industrial approaches for copy protection of microcontroller–based ECUs. Most approaches rely on cryptography and the possession of a secret key to distinguish between an original and a counterfeit ECUs; the assumption is that this key cannot be transferred to a counterfeit ECU. Hence, a counterfeit ECU will fail to complete subsequent cryptographic operations. The secure storage of keys can be implemented by tamper–resistant hardware modules [1]. Another possibility is to use a Physically Uncloneable Function (PUF) [17].

A. Microcontroller Security Fuse Bits

The typical automotive microcontroller (e.g., a MPC555) does not implement sophisticated security mechanisms to protect its internal flash memories from external access. The software stored on in this memory is usually protected by a fuse bit that – if unset – disables access to the flash memory. Technically, a fuse bit is wired with an enable signal for the memory in an AND gate.

B. Memory Encryption

Memory encryption techniques protect the contents of persistent or volatile memories and provide confidentiality on a micro-architectural level. The program code is stored as ciphertext and it is encrypted or decrypted when accessed. Therefore, the accessor has to present the correct key to store or retrieve meaningful data. For instance, modern Field Programmable Gate Arrays (FPGAs) store their programming file (i.e., the bitstream) in encrypted form. When loading the encrypted bitstream to the FPGA's switching fabric, it is decrypted on–the–fly.

Scrambling devices can render the memory contents useless for certain read out operations. Here, an encryption hardware module is installed between the Microcontroller Unit (MCU) and the flash memory. This effectively prevents the read out of the memory, for example via the OBD interface. The scrambled memory contents can only be decoded by a specific tool running on the maintenance computer [1]. The work in [27] suggests to integrate the encryption unit with a memory. Accesses to the memory when executing load and store instructions work transparently. This setup prevents external read out attempts, for example by connecting a device to the memory pins.

C. RFID Technology

Radio Frequency Identification (RFID) technology [8] is widely applied to combat counterfeiting. For this purpose, an RFID tag is attached to a physical object and thereby assigns this object a unique, virtual ID. The basic anti–counterfeiting protocol [24] uses PUF and works as follows: A built–in RFID reader periodically challenges the tags within its range and the tags respond accordingly. This response is verified by the reader using a database with responses which have been precomputed during an enrollment phase. Additional cryptographic means are applied to secure the single transmissions. This way, the reader determines the presence of ECUs in the vehicle's compartment.

[1] http://www.grautec.de/

D. Authentication Protocols

Authentication protocols are security protocols that enable receivers to distinguish, if a sender delivers an authentic or an corrupt service.

The authors of [23] propose a flexible authentication protocol that relies on a shared key between each sender and receiver pair. The protocol appends a truncated Message Authentication Code (MAC) to each message. By combining the MACs of several messages over time, the receiver determines a component's authenticity. A key insight is that the resulting delay between signing a message and its authentication are compensated by the safety mechanisms and the *inertia* of the controlled plant. Hence, the real–time system will tolerate wrong values until the messages are authenticated.

In [28], an authentication protocol based on [18] is presented. This protocol is adapted to time-triggered protocols (e.g., FlexRay on the automotive sector). Similarly to the protocol above, authentication is done with some delay.

E. Fallacies of the current approaches

The presented approaches are suitable to strengthen a computer system's confidentiality and integrity. Nevertheless, when considering the computer system of a vehicle, some fallacies become apparent.

Securing a system against memory read out is in vain. If the Return–on–Investment is high enough, even very sophisticated attacks to read out the memory are viable for attackers. First, the protection offered by fuse bits can be very weak. The author of [22] discusses non–invasive and invasive techniques to erase fuse bits. A non–invasive technique is to deliberately applying a power glitch or clock glitch to the memory. Invasive attacks include dismantling the chip's package and, for instance, to expose protection fuses to UV light. Second, hardware means for memory encryption are a potential attack goal. If the attacker can repeatedly retrieve ciphertexts from the memory, ciphertextonly attacks are possible. Moreover, side–channel attacks can reveal the keys used for the memory encryption. Security mechanisms targeted to prevent a memory read–out are not sufficient by itself. However, as memory protection erects a barrier for attackers, it should be *part of a defense–in–depth strategy*.

The RFID approach to check the integrity of the vehicle's computer system has the fallacy that tags and readers build their own authentication network. This network operates in parallel to the vehicle's real–time networks. If an RFID reader detect a near tag, the semantics is that the tag within spatial proximity, but this does not imply that the corresponding ECU participates actively in the real–time system, i.e., the control loop. It could have been replaced by a powerbox or a counterfeit ECU that substitutes its service.

Authentication protocols can effectively operate in the scope of a real–time network. They can implement the security requirements of a copy protection scheme, given that some cryptographic keys can securely be stored in an ECU. Drawbacks are that they are tedious to implement in a resource–constrained environment. Furthermore, key management and provision of new key material is a complex task that has to be addressed. Therefore, we argue that *application–specific knowledge should be part of the authentication process.*

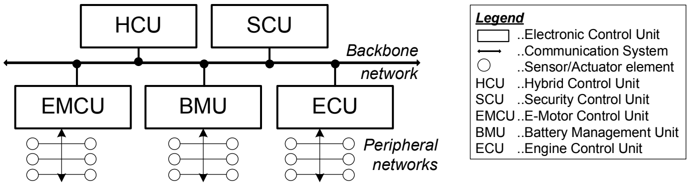
HCU
SCU
Backbone network
EMCU
BMU
ECU
Peripheral networks
Legend
▭ ..Electronic Control Unit
⟷ ..Communication System
○ ..Sensor/Actuator element
HCU ..Hybrid Control Unit
SCU ..Security Control Unit
EMCU ..E-Motor Control Unit
BMU ..Battery Management Unit
ECU ..Engine Control Unit

Fig. 2. ECU system architecture of a hybrid vehicle

IV. Authentication Architecture

In this section, we propose a generic security architecture that is capable to detect counterfeit ECUs. Our solution is particularly targeted at automotive setups, but could be applied to any other setups as well. The distinctive feature of our architecture is that it takes into account application–specific knowledge for the reliable authentication of ECUs.

A. Automotive System Architecture

Figure 2 depicts an exemplary ECU system architecture for a hybrid vehicle; it assembles different automotive subsystems around a backbone network (e.g., FlexRay, CAN). Each subsystem can itself embody a peripheral network structure (e.g., CAN, LIN). If an ECU exports information from a peripheral network into the backbone network, it acts as a gateway. For instance, the Battery Management Unit (BMU) senses the battery's current temperature and voltage, executes an State–of–Charge (SoC) determination algorithm, and forwards the result together with the temperature to the Hybrid Control Unit (HCU). We introduce in this setup a Security Control Unit (SCU) which performs the processing of the security–related information.

B. Authenticity Heartbeat Signal

Essentially, an *Authenticity Heartbeat Signal (AHS)* is an authenticated message between an ECU and the SCU. The contents of an AHS is the value of a process variable which is relevant for the subsystem under surveillance. This is, for instance, the maximum allowed dissipation from an electric vehicle's battery. An AHS has following properties:

- *Unidirectional*: the message flows from the control units to the SCU and there is no acknowledgment
- *Periodic*: it is transmitted periodically and it is timely (i.e., it is associated with deadline)
- *Authenticated*: It carries an authentication tag, e.g., a message digest, that is verifiable by a receiver
- *Unique*: Each message of the AHS is unique and therefore resistant to replay attacks
- *Plausible*: The contents of the message is credible, trustworthy, and acceptable;

An AHS can be emitted from any node in the system. If the node is located in a peripheral network, the signal has to be relayed by an ECU. The specification of this message

TABLE I
INTERPRETATION OF RESULTS AT THE SCU

	Original	Tuned	Counterfeit	Powerbox
Authentication	pass	pass	fail	fail
Plausibility check	pass	fail	pass	fail

abstracts over any particular communication protocol, because it might cross different networks according to our system model. The periodicity of an AHS can be much lower than that of the actual signal. This enables a tailoring between detection accuracy and resource consumption.

C. Processing at the Security Control Unit

The SCU is the terminal point for all AHS. It gathers all security–relevant information, performs a detection of malicious behavior, and reports to the user. For reasons of fault tolerance, it can be replicated. Basically, an SCU performs three tasks:

1) *Authentication*: The SCU authenticates the messages of an AHS to ensure their integrity of source and contents. Authentication is implemented by appropriate security protocols. Authentication enables a secure association between an AHS and an ECU. Each ECU has to securely store a root key in a non–volatile memory.
2) *Plausibility checking*: Next, the SCU checks, if the communicated process variable is plausible. This processing step requires application–level knowledge. Appropriate methods are for example investigated in the field of anomaly detection [2].
3) *Reporting*: Finally, the SCU reports to the user or a maintenance engineer, for example, using the dashboard.

Table I specifies the interpretation of the authentication and plausibility checks. If an attacker has induced modified parameters in the application image, the application will mostly likely diverge from the specified behavior. The plausibility check aims to detect patterns that indicate a deviation from the specified behavior in an AHS. The analysis of application–specific knowledge enables the detection of chip tuning. If an ECU fails to authenticate or omits expected messages of the AHS, the SCU can reason that this particular ECU is either missing, faulty, or unauthentic. According to our reasoning, unauthentic ECUs are also counterfeit. A powerbox will fail both checks: it is unauthentic and delivers a corrupt service.

D. Mapping to ACROSS service model

The ACROSS project defines a cross-domain reference architecture for embedded systems [14]. At its core, the project implements the ACROSS architecture as an Multi–Processor System–on–a–Chip (MPSoC) which facilitates the robust integration of several ECUs on a single silicon die. The architecture defines a set of security services that can be used to implement the proposed authentication architecture.

The *Periodic Messaging Service* enables per definition the first two properties of an AHS: unidirectionality and periodicity. The *Secure Group Communication Service* implements appropriate security protocols which provide the authenticity and uniqueness properties. The SCU is realized as an additional system component.

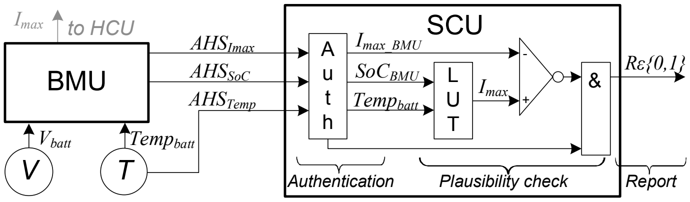

Fig. 3. AHS emission and processing at the SCU

TABLE II
MESSAGES IN THE BATTERY MANAGEMENT EXAMPLE

Name	Unit	Receiver	Sender	Rate [Hz]
$Temp_{batt}$	$°C$	BMU	Temperature sensor	100
V_{batt}	V	BMU	Volatage sensor	100
$AHS_{I_{max}}$	A	SCU	BMU	1
AHS_{Temp}	$°C$	SCU	Temperature sensor	1
AHS_{SoC}	%	SCU	BMU	1

V. EVALUATION

In a hybrid vehicle, the BMU is a key ECU to control the power delivered to the engines. For this purpose, it computes the maximum constraints $I_{max\_discharge}[A]$ and $I_{max\_charge}[A]$ for electro motor battery dissipation and supply. Usually, these values are precomputed and stored in two lookup tables – one for each entity – in the BMU. Depending on battery temperature $Temp_{batt}[°C]$ and SoC[%], the BMU forwards the according value for $I_{max\_discharge}[A]$ and $I_{max\_charge}[A]$. A tuner might modify the lookup tables to widen maximum charge and discharge limits. The gain would be to have more power readily available to consume, for instance to improve the vehicle's acceleration. The drawbacks are increased charge cycles and a decreased battery lifetime. These drawbacks might as a consequence discredit the vehicle's manufacturer and suppliers, and, in the worst case, cause the batteries to ignite.

Figure 3 displays a sample implementation to verify the authenticity of the $I_{max\_discharge}$ value in a hybrid vehicle. The BMU emits – in addition to its normal function – three AHSs for $I_{max\_discharge}$, SoC, and $Temp_{batt}$. These signals are emitted at a much lower rate than the signals for the control algorithm. In the first stage, the SCU authenticates all three signals. A delay in the computation – as for instance introduced by the time–delayed authentication technique – is not a problem, because the AHSs are not in the real–time path. However, an upper bound to the delay must be known to enable a correct reporting. Next, the plausibility is checked by using the same procedure (lookup tables) as implemented in the BMU. Finally, the SCU reports.

VI. RELATED WORK

The security in VANets is a hot research topic. Several publications investigate on security architectures for vehicles [16] [29] [5]. None of them lists copy protection or digital rights protection as a primary concern. Various books have been published on automotive security; [30] shortly discusses counterfeit ECUs. The paper in [3] proposes a security framework for vehicles. One objective of this framework is to facilitate *part authorization*. Contrarily to our work, it does not include application–specific knowledge in the authentication process.

VII. Conclusion

In this paper, we introduce copy protection for ECUs as a new application in industrial informatics. We strongly believe that mechanisms for IP protection will enable and sustain the newly emerging markets in the embedded systems field. As our main contribution, we proposed an authentication architecture based on the the notion of an Authenticity Heartbeat Signal. An AHS is an authenticated message carrying a relevant process variable of an ECU under surveillance. A special unit, namely the Security Control Unit, performs then a rigorous check of the system state with respect to malicious modifications. The output of the SCU can be interpreted as an electronic warranty seal that determines, if the system state corresponds to the manufacturer's specification. Therefore, the SCU has to carry application–specific knowledge. The usage of application–specific knowledge is a novel approach to implement copy protection. Current approaches focus rather on the research of 'pure' computer security mechanisms that rely on cryptography. Moreover, our solution does not interfere with the real–time properties of the target system, but it separates the authentication procedure from the processing of the control loop. We evaluated our concept for the battery management subsystem of a hybrid vehicle and thereby showed its feasibility. Future work is to link our work with AUTOSAR to make our research results available for the industrial uptake.

Acknowledgment

This document is based on the ACROSS project in the framework of the ARTEMIS Joint Undertaking. The work has been funded in part by the ARTEMIS Joint Undertaking and National Funding Agencies of Austria, Germany, Italy and France under the funding ID ARTEMIS-2009-1-100208. The responsibility for the content rests with the authors. The authors would like to thank Christian El–Salloum and Andreas Fleck for their support and the fruitful discussions.

References

[1] FIPS PUB 140-2. Security requirements for cryptographic modules. National Institute of Standards and Technology (NIST), 2002.

[2] Arindam Banerjee and Vipin Kumar. Anomaly Detection: A Survey. Technical report, ACM Computing Survey, September 2009.

[3] Hagai Bar-El. Intra-Vehicle Information Security Framework. In *Conference on Embedded Systems in Cars (ESCAR)*, 2009.

[4] Business Software Alliance (BSA). 2010 Global Piracy Study, 2011.

[5] M. Gerlach, A. Festag, T. Leinmüller, G. Goldacker, and C. Harsch. Security architecture for vehicular communication. In *4th International Workshop on Intelligent Transportation (WIT)*, 2007.

[6] Andreas Gissler and Jasmin Müller. Automotive After Sales 2015: Are you ready for the battle? Whitepaper, Arthur D. Little, 2008.

[7] John Humphrey and Olga Memedovic. The global automotive value chain: What prospects for upgrading by developing countries. Sectoral studies series, UNIDO, Strategic Research and Economics Branch, 2003.

[8] Ari Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[9] Roland Kammerer, Bernhard Frömel, and Armin Wasicek. Enhancing Security in CAN Systems using a Star Coupling Router. In *IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2012.

[10] Paul C. Kocher, Ruby B. Lee, Gary McGraw, Anand Raghunathan, and Srivaths Ravi. Security as a new dimension in embedded system design. In *Proceedings of the 41th Design Automation Conference, DAC*, pages 753–760. ACM, June 2004.

[11] Hans Löhr, Ahmad-Reza Sadeghi, and Marcel Winandy. Patterns for secure boot and secure storage in computer systems. In *International Conference on Availability, Reliability, and Security (ARES)*, 2010.

[12] Sean P. McAlinden and David J. Andrea. Estimating the new automotive value chain. Study CAR/ALTARUM 2002-07, Center for Automotive Research, Altarum Institute, 2002.

[13] Chris Murphy. Why Ford Just Became A Software Company. Online, November 2011.

[14] R. Obermaisser, H. Kopetz, and C. Paukovits. A cross-domain multiprocessor system-on-a-chip for embedded real-time systems. *IEEE Transactions on Industrial Informatics*, 6(4):pp. 548–567, 2010.

[15] Roman Obermaisser, Philipp Peti, Bernhard Huber, and Christian El Salloum. DECOS: An Integrated Time-Triggered Architecture. *”e&i journal (journal of the Austrian professional institution for electrical and information engineering)”*, 3:83–95, 2006.

[16] Panagiotis Papadimitratos, Levente Buttyan, Tamas Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and Jean-Pierre Hubaux. Secure Vehicular Communication Systems: Design and Architecture. *IEEE COMMUNICATIONS MAGAZINE*, 46(11):100–109, NOV 2008.

[17] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical One–Way Functions. *Science*, 297(5589):2026–2030, 2002.

[18] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. Spins: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, September 2002.

[19] Heinrich Prankl and Hermann Schaufler. Motortuning zur Leistungssteigerung an Traktoren. Bericht BLT053322, Francisco Josephinum Wieselburg, 2006.

[20] S. Roth. Informatisierung in der Automobilindustrie. In *Informatisierung der Arbeit Gesellschaft im Umbruch*, 2005.

[21] SAE. E/E Diagnostic Test Modes. Standard J1979, Vehicle E/E System Diagnostic Standards Committee, September 2010.

[22] Sergei P. Skorobogatov. Copy Protection in Modern Microcontrollers. Online, 2001. http://www.cl.cam.ac.uk/ sps32/mcu_lock.html.

[23] Christopher Szilagyi and Philip Koopman. Flexible Multicast Authentication for Time-Triggered Embedded Control Network Applications. In *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pages 165–174, 2009.

[24] Pim Tuyls and Lejla Batina. RFID-tags for Anti-Counterfeiting. In *Topics in Cryptology - CT-RSA*, volume 3860 of *LNCS*, 2006.

[25] Ingrid Verbauwhede and Patrick Schaumont. Design methods for security and trust. In *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, pages 1–6, April 2007.

[26] Henning Wallentowitz, Arndt Freialdenhoven, and Ingo Olschewski. *Strategien in der Automobilindustrie: Technologietrends und Marktentwicklungen*. Teubner Verlag / GWV Fachverlage GmbH, 2009.

[27] Armin Wasicek and Christian El Salloum. End-to-End Encryption in the TTSoC Architecture. In *Proceedings of the 3rd Workshop on Embedded Systems Security, ESWEEK'08, Atlanta, USA*, October 2008.

[28] Armin Wasicek and Christian El Salloum. Authentication in time–triggered systems using time-delayed release of keys. In *Proceedings of 14th IEEE International Symposium on Object/component/service-oriented Real-time distributed computing (ISORC)*, April 2011.

[29] M. Wolf and T. Gendrullis. Design, implementation, and evaluation of a vehicular hardware security module. In *14th International Conference on Information Security and Cryptology*, 2011.

[30] Marko Wolf. *Security Engineering for Vehicular IT Systems*. Vieweg+Teubner, 2009.