2018 IEEE International Conference on Smart Computing

Analyses of secure automotive communication protocols and their impact on vehicles life-cycle

Dario Stabili, Luca Ferretti and Mirco Marchetti
University of Modena and Reggio Emilia, Italy
Email: {dario.stabili, luca.ferretti, mirco.marchetti}@unimore.it

***Abstract*—Modern vehicles are complex cyber physical systems where communication protocols designed for physically isolated networks are now employed to connect Internet-enabled devices. This unforeseen increase in connectivity creates novel attack surfaces and exposes safety-critical functions of the vehicle to cyber attacks. As standard security solutions are not applicable to vehicles due to resource constraints and compatibility issues, research is proposing tailored approaches to cope with existing systems and to design next generation vehicles. In this paper we focus on solutions based on cryptographic protocols to protect in-vehicle communications and prevent unauthorized manipulation of the vehicle behaviors. The existing approach is to consider vehicles as monolithic systems and evaluate performance and costs of the proposed solutions without considering the complex life-cycle of automotive components and the multifaceted automotive ecosystem that includes a large number of actors. The main contribution of this paper is a study of the impact of security solutions by considering vehicles life-cycle. We model existing proposals and highlight their impacts on vehicles production and maintenance operations by taking into consideration interactions among multiple players, and we give insights on the requirements of the architectures that are necessary to deploy secure intra-vehicular protocols.**

***Index Terms*—automotive, security, CAN, integrity, authenticity, life-cycle**

I. Introduction

The increasing adoption of advanced driver assistance systems and infotainment solutions, often connected to the public Internet, makes modern vehicles similar to mobile networks of computing devices. These technologies can pave the way to novel business models and can increase the vehicles' safety, such as the European eCall initiative [1]. However, they inevitably expose novel attack surfaces that cyber criminals can exploit to affect the safety of the whole vehicle [2], especially due to the lack of proper security countermeasures in the communication protocols that compose the internal network of a vehicle [3].

All proposals striving to improve the security of vehicle networks must tackle many non-trivial issues caused by the severe constraints of communication buses and embedded computing elements deployed in modern vehicles. The highly competitive automotive market imposes the adoption of low-costs solutions, in which computing elements are micro-controllers (ECUs) with low-end computational capabilities and the most popular legacy network protocol (the Controller Area Network protocol, or CAN) is characterized by short messages and low bandwidths that are incompatible with standard security solutions commonly applied to the TCP/IP stack.

Previous works in the literature propose lightweight cryptographic protocols and in-vehicle architectures for guaranteeing data integrity and authenticity that are effective in preventing many classes of attacks [4], [5], [6], [7], [8], e.g., by adding some sort of message authentication mechanisms that allows well-behaving ECUs to detect and reject illicit messages injected in the CAN bus by an attacker. Another class of defense solutions follows the approach of intrusion detection rather than intrusion prevention, and includes algorithms for the detection of malicious messages within the internal networks [9], [10], [11], [12], [13], [14], [15], [16]. These proposals are orthogonal to secure cryptographic protocols and can be added to existing vehicular systems with few modifications on the pre-existing architecture.

In this paper, we focus on cryptographic protocols for in-vehicle network security and provide an additional analyses of existing proposals based on the vehicles life-cycle. Existing proposals usually evaluate security guarantees, performance and costs of security protocols deployed in a resource-constrained system such as the CAN network of a vehicle. As an example, a typical trade-off involves security solutions based on asymmetric cryptography, that achieve high security levels thanks to fine-grained key distribution, but require additional expensive hardware devices such as Hardware Security Modules (HSM). However, most proposals consider vehicles as monolithic systems and do not discuss the issues of properly deploying and managing all the required cryptographic materials on the vehicle components. Vehicles are complex systems that involve many actors since early design, production and assembly phases, where all processes are extremely optimized to minimize costs.

In this paper, we show that the adoption of a security solution based on cryptography affects many of these processes, and that different security solutions might have different impacts on the vehicle life cycle. We argue that analyses that do not consider the entire life-cycle of automotive vehicles cannot evaluate the feasibility of the proposed security solutions in the real world. As an example, a vehicle manufacturer must consider that adopting secure cryptographic protocols for communications requires a correct management of the cryptographic keys, such as installation within the production lines, secure storage during the vehicles life-cycle, and possibly sharing this critical information with suppliers, business

978-1-5386-4705-9/18/$31.00 ©2018 IEEE
DOI 10.1109/SMARTCOMP.2018.00045

partners and other legitimate third parties. We claim that secure solutions for vehicular communications are strictly tied to the entire life-cycle of vehicles. Different solutions might have different impacts on global costs and, in the worst case, some of them might prove too costly or unfeasible to deploy in real-scenarios.

The paper is organized as following: Section II describes vehicles system and threat models; Section III describes security proposals for in-vehicle network protocols; Section IV models the life-cycle and highlights the main requirements of a scalable security architecture; Section V outlines conclusions and future work.

II. Vehicles system and threat models

Modern vehicles are complex cyber physical systems that include many microcontrollers called Electronic Control Units (ECUs) connected via an intra-vehicular network (for short, in-vehicle network or vehicle network). ECUs implement all the control logic of software-driven features, including many safety-critical functions such as the steering and braking, and many other less-critical features such as the air conditioning and the infotainment system. ECUs are physically connected to the vehicle network, and communicate with each other through specialized protocols that satisfy all functional requirements, such as real-time communications and message prioritization, while guaranteeing the lowest possible production cost. Multiple networks can coexist within the same vehicle to isolate different functionalities, but a single network usually serves multiple purposes to optimize costs. As an example, entry-level vehicles might be provided with only one network that connects ECUs involved in safety-critical features and other ECUs implementing the infotainment system. This design choice is one of the causes of recent successful cyber attacks to vehicle networks, where attackers exploit vulnerable connectivity interfaces exposed by the infotainment system to access safety-critical functions [17].

In this paper we focus on the *Controller Area Network* (CAN) protocol for the exchange of messages over a shared bus in real-time [18], that is the most popular protocol for vehicle communications. The CAN protocol enables communications via a message-oriented paradigm among groups of ECUs by using the CAN ID field. At setup time, each ECU is associated to one or multiple CAN IDs. To specify a recipient, an ECU sending data sets the CAN ID field of the message: since CAN uses a broadcast channel all ECUs receive data, but only those associated to the CAN ID set in the message accept it. Similarly to most protocols for cyber physical systems (e.g., ModBus for SCADA applications), CAN does not include any cyber security functionality because its original design assumed physically isolation of the communication buses and the absence of external interfaces. As a result, any ECU can inject arbitrary messages on the CAN bus, and can set any CAN ID in the output data without the possibility of enforcing any authorization policy within the network. To protect against cyber attacks, research is putting effort in extending CAN.

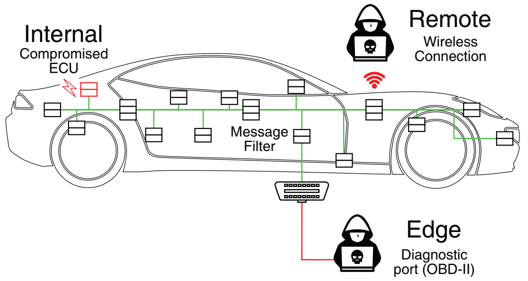

Fig. 1: Attack surfaces of a modern vehicle

In the following, we outline how adversaries can attack in-vehicles networks. Then, in Section III we discuss existing proposals to extend CAN with additional security guarantees.

Cyber attacks to modern vehicles aim at modifying the dynamic of the vehicle or a given safety-relevant feature through the multi-step process described below: *(a)* gaining a privileged access to the vehicle network; *(b)* acquiring information about the protocol adopted in the network; *(c)* injecting malicious messages in the vehicle network to manipulate the information processed by a target ECU.

(a) As represented in Figure 1, a modern vehicle offers *internal*, *edge* and *remote* attack surfaces:

- the *internal* surface represents any type of direct access to a segment of the vehicle network via a compromised ECU or by physically hijacking the twisted pair of the CAN bus. In this paper we consider a powerful attacker that managed to gain full control of an ECU connected to the same CAN bus segment of the target ECU. This scenario gives the attacker the possibility to access both volatile and persistent memory, and to execute arbitrary code on the compromised ECU [7];
- the *edge* surface represents cabled interfaces originally exposed within the vehicle, usually for diagnostic reasons, such as the On-Board Diagnostics (OBD-II) port of CAN. We assume that the manufacturer assigned limited privileges to this interface, such as read-only access to the messages exchanged on the network. Note that if the manufacturer exposes directly the CAN bus on this interface, then this attack surface falls back to the *internal* case;
- the *remote* surface represents external interfaces provided by wireless communication services, such as Bluetooth or WiFi, possibly connected to the Internet. Breaking security mechanisms of these interfaces, or of related exposed applications, might allow attackers to obtain privileged access to the ECUs that implement them and escalate to *internal* access privileges.

(b) A successful targeted attack to the vehicle network requires detailed knowledge about the syntax and semantic of messages transmitted over the CAN bus. Although some aspects of the syntax of vehicle protocols are mandated by public standards, each manufacturer adopts different choices to encode message identifiers and signals within the message

payloads. As an example, although the CAN standard is public, each vehicle uses different CAN IDs and represent information by using different encoding techniques (e.g., the engine revolutions per minute are represented through different binary representations and scales). This information is kept confidential among manufactures and suppliers and is not released in the public domain, thus the attacker has to apply reverse-engineering techniques to infer these information from the CAN traffic [19]. If attacks target a specific vehicle model, the attacker can run this preliminary phase as an offline operation on an identical vehicle at his disposal. If the attacker targets a wide-range of vehicles, reverse-engineering is executed by sniffing the traffic within the vehicle network [17]. As an example, *fuzzing attacks* have been proved effective to gain useful insights about the vehicle network topology and to map the connected ECUs [20].

(c) A compromised ECU connected to the CAN network is able to operate many types of safety-critical attacks. Known examples include: *ECU shutdown* [21] and *ECU impersonation* [16] (also, *masquerade* attack) to activate the *bus-off state* mode, shut down an ECU and mimic its behavior by sending properly forged payloads to subvert the vehicle functions; *denial-of-service* or *replay* that are able to manipulate the normal behavior of the vehicle dynamic. All of these attacks can be executed from different ECUs of the vehicle. However, we note that certain scenarios might require that a specific ECU processes some fake data injected in the network, while others only require that any ECU accepts and processes the forged data. As an example, a typical trade-off is to design security measures that isolate ECUs of different segments of the network, such as powertrain, body, infotainment or a group identified by a CAN ID.

The proposed model is an engineered threat model based on attacks already analyzed by the literature. In this paper, we consider an additional variant, that is attacks that are based on the access on multiple vehicles and on the different security guarantees that vehicles should guarantee in this scenario. This is a characteristic that we consider when analyzing security protocols and architectures for the vehicles life-cycle. When attacking a vehicle, an adversary might use an extension of the multi-step approach described above by leveraging the fact that multiple vehicles are produced through an industrial serialized methodology. If multiple vehicles share some secret information used in security protocols, the attacker can extract information about a target vehicle from another vehicle, such as one of the same model or producer. Intuitively, this might ease the reverse-engineering process described in phase (b), but we observe that it also allows to infer information about security protocols deployed in the vehicle and potentially of secret cryptographic keys. As an example, if different vehicles use the same cryptographic keys, even if they use them to protect ECUs communications with high granularity, then the adversary can obtaining these secret keys from a similar vehicle at his disposal, on which he has *internal* access, and use them to compromise the security of the target vehicle, on which he only has *external* access. The security guarantee the we define as *inter-vehicle security independence* identifies a security solution where vehicles are protected against these attacks, that is, an attacker cannot gain any advantage in accessing secret information stored in a vehicle by attacking any other vehicle.

We observe that this security guarantee is strictly tied to security measures that protect devices against physical access, that is, white box attacks. If we assume that an adversary can have the same advantage in attacking a vehicle by having physical access to any other vehicle, than deploying white-box security defenses such as temper-resistant hardware modules or white-box cryptography is of paramount importance. However, if it is possible to distribute independent keys for each vehicle and to guarantee inter-vehicle security independence, then car manufacturers might achieve similar levels of security without white-box defenses. Indeed, attacking a target vehicle would require the adversary to physically access that very same vehicle, that is a much weaker security assumption.

III. Security solutions for intra-vehicular networks

In this section we analyze existing proposals for the design of secure vehicle networks and we model the common traits that we use to identify their impact when considering the vehicle life-cycle. We identify two main challenges in extending CAN with security guarantees: the design of a secure transport protocol to authenticate data (Section III-A) and the distribution of the required cryptographic material (e.g., shared keys) to all interested ECUs (Section III-B).

A. Secure transport protocol for CAN

Standard security solutions to guarantee data integrity and authenticity usually compute and concatenate to the message a tag generated with a Message Authentication Code (MAC) protocol. This approach guarantees that any illegitimate modification on the data (i.e., by someone that does not know the secret key) can be detected by the recipient. However, the CAN protocol uses fixed small-size packets and low bandwidth channels that make it unfeasible to attach a MAC to each message. The simplest way to deploy efficient and secure authenticated protocols would be to use intra-vehicular network protocols that are more flexible than CAN, such as the the CAN+ extension. These protocols support larger messages and allow to associate MAC tags to messages in a more standard fashion [4]. However, car manufacturers are not prone to adopt them due to the increased costs. We identify two approaches for extending the CAN protocol with guarantees of data authenticity. The first approach is to let each ECU produce additional CAN messages to transmit MAC tags. This approach offers the best security, however it also introduces huge network overhead that makes it unfeasible in most automotive networks. The second approach represents a trade-off between performance and security. It requires each ECU to authenticate batch of messages by using a single MAC tag, to fragment it and to send the fragments within CAN headers of the messages [5]. The main disadvantage of the approach is that a recipient can verify authenticity of data only once every few messages: since CAN is a real-time protocol and ECUs process messages as soon as they are received, an attacker can inject malicious messages without being detected for a certain time interval. A second disadvantage is that MAC fragments are transmitted within existing fields of CAN headers, such as the CRC, possibly affecting existing protocols design.

Guaranteeing protection against replay and reflective attacks also requires additional design choices at the application and architectural levels, as typical for standard communication protocols. In the context of vehicle networks, proposals exist based on centralized time-servers [6], distributed counters [7] and key-derivation approaches [8]. However, we highlight that these solutions do not impact architectural design choices because they do not involve the distribution of additional persistent cryptographic material. Thus, the adoption of any of these solutions is orthogonal to the analysis proposed in this paper.

B. Intra-vehicular ECU keys distribution

We consider three types of approaches to distribute key material to the ECUs: *pre-shared ECU keys*, *in-vehicle key distribution centers* and *certificate-based key authentication*.

Pre-shared ECU keys. The first approach to deploy secure protocols is to install symmetric keys in the ECUs persistent storage, where the same key is deployed within all ECUs that must communicate. Different key distribution strategies can be adopted to obtain different trade-offs in terms of storage overhead and security guarantees. As an example, a master key could be stored in all ECUs or multiple group keys could be selectively stored in ECUs depending on their roles within the vehicle (e.g., their associated CAN IDs).

In-vehicle key distribution centers. The second approach based on symmetric cryptography requires to install additional ECUs within the vehicle that act as Key Distribution Centers (KDCs) [22], [23], [6]. Each KDC knows the secret keys of a subset of the ECUs and releases the due session keys to enable pair-wise or group communications. As in standard secure communication protocols, this approach enables good security guarantees, low storage overhead and easier management of persistent keys distribution. In vehicle networks, it has the disadvantage of introducing additional costs due to the additional ECUs and some network overhead.

Certificate-based key authentication. The last approach is to adopt primitives based on asymmetric cryptography, such as digital signatures and certificates, to deploy solutions that are similar to the Internet Public Key Infrastructure (PKI) architecture [24]. Each ECU is configured with a list of trusted Certification Authorities (CAs) and stores a key pair signed by one of the trusted CAs. ECUs can communicate with each other by exchanging symmetric session keys by using key exchange protocols. An optional variant of this architecture is to also include a specialized ECU within the vehicle that can revoke invalid certificates. Despite great advantages in terms of security and easier management, most ECUs have tight resource constraints and do not support asymmetric cryptography. Thus, deploying these solutions require the addition of specialized Hardware Security Modules with increased costs.

IV. Security architectures for vehicles life-cycle

We model the vehicle life-cycle by considering three actors:

- *producer*: the company that designs and produces the vehicle;
- *OEM*: a company that produces vehicle components. It is responsible for providing maintenance, assistance and replacement parts during the vehicle life-cycle, that includes software in case of electronic components;
- *owner*: a person that buys the vehicle and uses it;
- *maintainer*: a company or a private that operates maintenance on the vehicle.

We represent the vehicle life-cycle by using a finite-state machine model, where each state represents a phase in the life-cycle of the vehicle. Each state is also associated to an actor, that we call the *authoritative* actor, that has exclusive access to the vehicle during the corresponding phase. The actor can cause a transition to another state of the life-cycle, possibly passing the authority over the vehicle to another actor. The model assumes that the actor associated to a state has physical access to the vehicle. Note that this does not imply that the actor has full access over the vehicle, as this might be limited by his knowledge of the vehicle components and his technical capabilities. As an example, we can assume that a maintainer can accomplish advanced repair operations, but we should assume that the owner might only be able to drive the vehicle.

We describe the details of the model by referring to Figure 2, that shows the different states of the model and highlights the authoritative actor for each phase. The first state of the diagram is the *design* of the vehicular network, where the specifications of each ECU that will be deployed in the vehicle are defined. The authoritative actor of this state is the producer. The results of the *design* state are the software specifications for the ECUs to be deployed in the vehicle, which are the input of the *production* phase, where the software to be installed on the ECUs is coded. The ECUs hardware is bought from an external hardware producer directly by the OEM, which is the authoritative actor of this state. The producer can designate different OEMs to produce part of the vehicle components or a single OEM to produce all the components. The third state of the vehicle life-cycle is the *assembling*, where the ECUs with the software already installed are delivered by the OEM and assembled together with the mechanical parts of the vehicle. The authoritative actor of this step is the producer. After the assembling state the vehicle is available on the market, and after it is sold the *operational* state begins. The vehicle in this state is considered fully operational and at the disposal of the owner, which is the authoritative actor of this step. During the operational phase both ordinary or extra-ordinary service operations are required, thus the maintainer gains control over the vehicle and the *maintenance* state

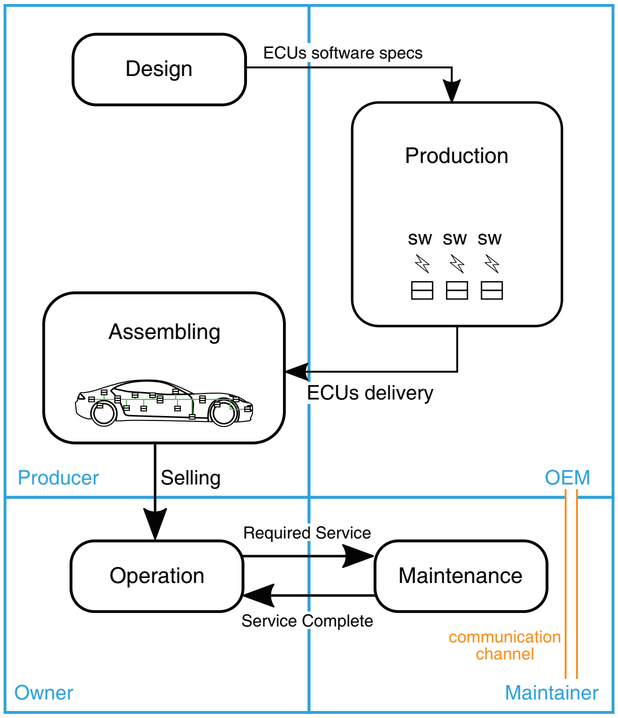

Fig. 2: Vehicle life-cycle

is entered. Maintainers are the authoritative actors of this step and act as intermediaries between the vehicle and the OEM, requiring special access to the vehicle components, such as their configuration, and privileged access to parts of the vehicle if needed. Once the service is completed, the vehicle returns in the *operational* state. Multiple transitions between the *operational* and *maintenance* states can occur during the normal vehicle life-cycle.

Supporting security solutions for vehicle networks requires the design of an architecture that allow the distribution of the due cryptographic material by actors involved in the life-cycle. In particular, the main non-trivial design choices regard the *production* and *assembling* phases implemented by the *OEM* and the *producer*, that must generate and share inter-dependent secret information with the aim of obtaining the best security guarantees. Depending on the designed architecture, *maintainers* might have to interact in different ways to support their customers. In this paper we are not interested in the details of each protocol required within the architecture, such as information sharing protocols among players or secure management of cryptographic keys, but we highlight peculiar traits of the architecture that depend on the adopted security solutions for intra-vehicle communications. Moreover, we discuss how these architectures can guarantee the *inter-vehicle security independence* as discussed in Section II. In the following, we discuss existing security protocols by distinguishing them in *pre-shared ECU keys* (Section IV-A), *in-vehicle key distribution centers* (Section IV-B) and *certificate-based key authentication* (Section IV-C), that represent families of in-vehicle key distribution strategies as described in Section III.

A. Pre-shared ECU keys

Deploying a security protocol based on pre-shared ECU keys requires multiple OEMs to share cryptographic keys with each other and with the producer. This requirement is mandatory for any level of granularity, such as using a global master key, group keys or pair-wise keys. However, some design choice might be influenced by key granularity. If multiple OEMs produce components that communicate with each other (e.g., associated to the same CAN IDs), then the producer is the only player that has a global view of the system and that can take care of key generation and distribution. In this case, the producer can decide to choose either a master, group or pair-wise keys and distribute them to OEMs accordingly. Otherwise, if an OEM has exclusive responsibility for a certain group and a group key strategy is used, then he can autonomously generate and manage the secret key. However, the main issue in these architectures is the generation and deployment of different cryptographic keys for different vehicles due to the reconciliation of the keys at the assembly phase. To enable inter-vehicle security independence, OEMs must install different keys on each component and keep track of the components that share the same keys. Then, the producer should handle the reconciliation of all components that share secret keys to assemble them in the same vehicle. Although this kind of management seems theoretically feasible, it puts a lot of burden on both the OEMs and the producer. Moreover, since components are not interchangeable, it introduces complex issues in case of failures. As a result, solutions based on pre-shared ECU keys do not seem a viable design choice to guarantee inter-vehicle security independence.

B. In-vehicle key distribution centers

Deploying a security protocol based on in-vehicle key distribution centers require the OEMs and the producer to share pair-wise cryptographic keys between "normal" ECUs and the "special" ECU that implements the KDC (also, KDC-ECU). This class of solutions might be implemented by using different design strategies. To implement an efficient and scalable architecture, we propose to let the OEM of the KDC-ECU monitor the assembly phase. As an example, the producer could maintain the production and management of the KDC-ECU in-house. By considering this assumption, the management of cryptographic keys can be implemented as following. We consider that the OEM received orders by the producer for a certain amount of components. The main objective is that each component stores secure cryptographic material that allows it to communicate with the KDC. Thus, at flashing time the OEM generates a random keys (or pseudo-random, if it uses a key derivation function) and installs them in the ECU. Then, the OEM sends the ECUs together with keys to the producer. Before the assembly phase, the KDC-ECU must contain the keys of all the ECUs that will be installed in the same vehicle. This operation seems feasible because all dependencies are resolved in the assembly phase. However, this architecture might require additional efforts to deploy maintenance operations. In case of ECU failures, substituting an ECU either requires: to obtain a new ECU that stores the same key of the failed one; to update the KDC with the key of the new ECU. Either design choice could be deployed with some effort, although the second option, that would require an update of the KDC-ECU persistent memory, seems more convenient. Indeed, requiring OEMs to flash a single ECU on-demand might be expensive.

C. Certificate-based key authentication

Deploying security protocols based on asymmetric cryptography enables the application of an operation flow that is similar to that of a standard PKI. We consider the following design choices. We assume that each OEM generates a certain number of secret keys and for each one produces a Certificate Sign Request (CSR). All CSRs are issued to the producer, that approves them and returns the due certificates. In each ECU, the OEM installs a secret key, the associated certificate and the public key of the producer. The software installed by the ECU will establish connections with ECUs that can produce certificates signed by the installed producer public key. This architecture represents an efficient approach to install the due cryptographic material in the ECUs, and has the great advantage of not distributing secret keys outside OEMs and outside a single ECU. However, it does not seem able to guarantee inter-vehicle security independence. Implementing this security guarantee would require the producer to use a different certificate for each vehicle and to sign the CSRs of the OEMs accordingly. Then, the producer would have to reconcile ECUS as described for the pre-shared ECU keys approach, that seems an unfeasible task. As a result, to obtain inter-vehicle security independence, the introduction of a centralized point of control that allows to define authorization policy on a per-vehicle basis at assembly time is not an optional choice, even when asymmetric cryptography is used.

V. Conclusions

In this paper we analyzed solutions for secure communications applied to intra-vehicular networks and Electronic Control Units connected to the CAN bus. We proposed a systematic analysis of typical attacks to modern vehicles and to the related solutions. In particular, we focused on their effects on the complete life-cycle of a vehicle, including components production, assembling and maintenance operations. Our analyses show that many solutions that may appear efficient and secure when deployed on a single vehicle, present severe disadvantages when deployed at scale. Results of our analyses show that solutions based on pre-shared symmetric secrets complicate the management and maintenance of the vehicle. Moreover, solutions that are not based on a centralized point of control managed by the car manufacturer, either based on symmetric or asymmetric cryptography, are not able to offer the desired security guarantees.

References

[1] European Parliament. Regulation (eu) 2015/758 concerning type-approval requirements for the deployment of the ecall in-vehicle system. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32015R0758

[2] Keen Security Lab of Tencent. (2016) Car hacking research: Remote attack tesla motors. [Online]. Available: http://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *Proc. 31st IEEE Int'l Symp. Security and Privacy*, 2010.

[4] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth: A simple, backward compatible broadcast authentication protocol for can bus," in *Proc. ECRYPT Workshop Lightweight Cryptography*, 2011.

[5] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," *Proc. IEEE 68th Conf. Vehicular Technology*, 2008.

[6] O. Hartkopp, C. Reuber, and R. Schilling, "MaCAN - Message Authenticated CAN," *10th Int'l Conf. Embedded Security in Cars*, 2012.

[7] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-Security for the Controller Area Network (CAN) Communication Protocol," in *Proc. 2012 Int'l Conf. Cyber Security*.

[8] A. I. Radu and F. D. Garcia, "Leia: a lightweight authentication protocol for can," in *21st European Symp. Research in Computer Security*, 2016.

[9] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proc. IEEE 2nd Int'l Forum Research and Technologies for Society and Industry Leveraging a better tomorrow*, 2016.

[10] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *Proc. 28th IEEE Symp. Intelligent Vehicle*, 2017.

[11] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *Proc. 2017 Int'l Annual Conf. Infrastructures for Energy and ICT*.

[12] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," *Proc. 2011 IEEE Symp. Intelligent Vehicles*.

[13] K.-T. Cho, K. G. Shin, and T. Park, "CPS Approach to Checking Norm Operation of a Brake-by-Wire System," in *Proc. Sixth ACM/IEEE Int'l Conf. Cyber-Physical Systems*, 2015.

[14] M. J. Kang and J. W. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *Proc. 83rd IEEE Int'l Conf. Vehicular Technology*, 2016.

[15] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *Proc. 2015 World Congress Industrial Control Systems Security*, 2015.

[16] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Symp. Security*, 2016.

[17] C. Valasek and C. Miller. (2014) Car Hacking: For Poories. [Online]. Available: http://illmatics.com/car_hacking_poories.pdf

[18] R. B. GmbH. (1991) Can specification version 2.0. [Online]. Available: https://www.kvaser.com/software/7330130980914/V1/can2spec.pdf

[19] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown can bus networks," *Vehicular Communications*, vol. 9, 2017.

[20] H. Lee, K. Choi, K. Chung, J. Kim, and K. Yim, "Fuzzing can packets into automobiles," in *Proc. IEEE 29th Int'l Conf. Advanced Information Networking and Applications*, 2015.

[21] A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack against automotive networks," in *Proc. 14th Int'l Conf. Detection of Intrusions and Malware, and Vulnerability Assessment*, 2017.

[22] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka, and H. Imai, "New Attestation-Based Security Architecture for In-vehicle Communication," in *Proc. 2008 IEEE Int'l Conf. Global Communications*.

[23] 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'l Conf. Cryptology and Network Security*, 2012.

[24] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Security in Automotive Networks: Lightweight Authentication and Authorization," *ACM Trans. Design Automation of Electronic Systems*, vol. 22, no. 2, 2017.