Combining Safety and Security in Autonomous Cars Using Blockchain Technologies

Lucas Davi, Denis Hatebur, Maritta Heisel, and Roman Wirtz

University of Duisburg-Essen, Duisburg, Germany
firstname.lastname@uni-due.de

Abstract. Modern cars increasingly deploy complex software systems consisting of millions of lines of code that may be subject to cyber attacks. An infamous example is the Jeep hack which allowed an attacker to remotely control the car engine by just exploiting a software bug in the infotainment system. The digitalization and connectivity of modern cars demands a rethinking of car safety as security breaches now affect the driver's safety. To address the new threat landscape, we develop a novel concept that simultaneously addresses both car safety and security based on the arising blockchain technology, which we mainly exploit to ensure integrity. Previous related work exploited the blockchain for the purpose of forensics, where vehicle data is stored on an externally shared ledger that is accessible by authorized third parties. However, those approaches cannot ensure integrity of information used by the vehicle's components. In contrast, we propose a blockchain-based architecture based on a shared ledger inside the car, where each ECU can act as a miner and shares its information with other ECUs. The architecture does not only improve the integrity of information for forensics. Some algorithms, e.g. the recognition of dangerous situations, are adaptive and can be improved using for example sensor data. Using our architecture, we ensure that those algorithms only take verified and correct information as input.

Keywords: Blockchain · Safety · Security · Autonomous vehicles

1 Introduction

Modern vehicles and especially next-generation autonomous cars increasingly face both security and safety challenges. On the one hand, car safety is crucial to ensure that a car functions correctly under various environmental circumstances. In particular, driving assistance systems help reducing the risk of a crash, e.g., the electronic stability program (ESP) significantly improves stability of the car, or recent automatic breaking systems reduce the risk of a collision by automatically braking the car. On the other hand, the increasing complexity and diverse features of a modern car require a vast amount of software and availability of communication channels to the Internet. This dramatically shifts the threat model as software running on a car is now interfacing with the Internet, thereby opening a new door to attackers. That is, similar to modern PC and

mobile systems, attackers can gain access to a remote channel to hijack modern automobiles. As one infamous example, researchers have demonstrated that a software bug in the Jeep infotainment system allowed them to remotely exploit the bug to threaten the driver's safety by gaining control on car internals such as engine, wipers, and fan [1]. This new interplay between security and safety aspects demands for a hybrid framework that considers both security and safety requirements. For instance, when adding new driver assistance technologies to increase the safety of the car, we also need to make sure that the software running on a car cannot be compromised to disable safety components.

In this paper, we devise a novel approach to develop such a framework based on blockchain technologies. In general, blockchain systems are mainly deployed to ensure secure and tamper-proof records of transactions. These transactions are typically exchanges of cryptocurrency (Bitcoin) or execution of the so-called smart contracts (Ethereum). The blockchain offers decentralized storage (i.e., a distributed ledger) to guarantee integrity of the transactions and non-repudiation. Further, and probably most importantly, it removes the need for a trusted third party (e.g., a bank). On the other hand, we observe in this paper that the blockchain features properties that are either similar to or can be exploited for safety requirements. First, the security of blockchain systems relies on replicating the current state of the blockchain on each client participating in the network. This introduced redundancy is similar to m-out-of-n (MooN) systems in the safety domain which ensure the functionality of the systems as long as m out of n systems are functioning correctly. Second, blockchain systems are based on consensus or majority voting algorithms to accept a newly proposed state (in form of a new block). Again, this is similar to safety systems based on modular redundancy which perform a majority voting to produce a single output. Furthermore, recording safety-relevant transactions in a decentralized fashion enables reliable and fast verification of accidents (forensics). This is similar to a blackbox used in airplanes. In the context of automobiles, recall the recent Tesla accident where it was unclear for several weeks whether the car was driving in autopilot mode before the accident occurred [2]. By manipulating recorded data, it is possible to blame innocent stakeholders. Therefore, integrity of recorded data is of high importance.

When mapping the aforementioned considerations to modern automobiles, the decentralized blockchain storage allows to securely record the transactions occurring inside a car and from a car to its environment (e.g., other cars or the infrastructure). Transactions inside a car are (i) messages sent between the different components - typically electronic control units (ECUs) - of a car, and (ii) state information of the car. When stored on the blockchain, every component participating in the internal network is able to inspect previous states and messages to decide whether the next state represents a legitimate state change. Basic example policies are (1) to not turn off the engine while the car is in driving state, (2) to not turn off the ESP if the designated button has not been physically pressed by the driver, i.e., this ensures that compromised software does

not broadcast a malicious message to turnoff the ESP, or (3) to detect abnormal state changes from the environment, e.g., sensors return mismatching results.

In fact, recent proposals leverage blockchain technologies to enable forensic investigation of autonomous cars [3,4]. However, these proposals target a higher abstraction layer, whereas we consider the layer of CAN messages to develop a framework that covers both safety and security requirements inside a car.

Deploying a blockchain framework to ensure security and safety of a modern car involves many challenges. First, there is a vast amount of messages constantly processed by the different ECUs in the car (cf. Section 2). Processing each of these messages in the blockchain would raise significant scalability problems due to performance reasons. Hence, the approach has to provide filter mechanisms for those messages that are relevant for the safety and security of the car. Second, there currently exists no framework that would allow us to determine whether the subsequent state is benign. As such, the next challenge to address involves new consensus algorithms that allow ECUs to make a decision on whether accepting or rejecting the next state. Finally, for the case of forensic-based validations, we need to develop a robust method to retrieve and process the current state of the blockchain. Our framework forms the basis for addressing those challenges by providing a blockchain-based architecture for an application inside cars.

The remainder of this paper is structured as follows: In Section 2, we introduce fundamentals on which our work is based, followed by a threat and system model in Section 3. We describe our blockchain-based architecture in Section 4 which is our main contribution. In Section 5, we discuss the results of an implementation of our architecture, and in Section 6 we discuss related work. Section 7 provides a summary of the paper and an outlook for future research directions.

2 Background

In this section, we introduce blockchains on which we built our architecture, followed by a description of a threat and system model stating assumptions on which we rely in the following.

2.1 Blockchain

Blockchain systems have become very popular over the last few years. The most famous examples of such systems are Bitcoin [5] and Ethereum [6]. The former allows anonymous transfer of cryptocurrency (Bitcoin) without requiring a trusted third party. Ethereum goes one step further than Bitcoin as it is not limited to transfer of cryptocurrency, but allows the execution of arbitrary computer programs (called smart contracts) on the blockchain. Prominent targets for smart contracts are crowdfunding, supply chain, decentralized autonomous organizations (DAOs), or micro-insurances which perform automated claim processing, thereby reducing the operating costs of insurance companies.

Abstractly speaking, the blockchain allows a secure, decentralized storage of blocks, where each block usually contains a limited number of transactions. Each

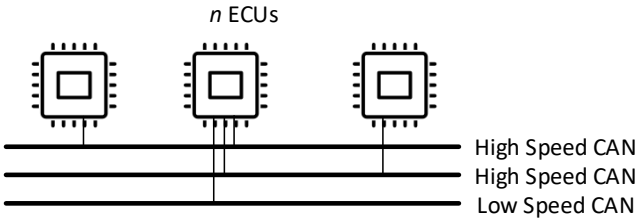

Fig. 1. Architecture Overview

block is cryptographically chained to its previous block by embedding the hash value of the previous block, thereby ensuring the integrity of the blockchain. As long as the majority of the network is not compromised by an attacker, the blockchain can prevent double-spending of coins and reversing of transactions. Blockchain systems are based on a consensus algorithm to accept a newly proposed block. The most well-known systems like Bitcoin and Ethereum deploy the so-called *proof-of-work (PoW)* consensus algorithm in which each node that participates in the so-called mining process must solve a mathematical puzzle and gets rewarded if it solves the puzzle first. Since PoW requires a lot of computational power and induces a high consumption of energy, *proof-of-stake (PoS)* algorithms have been proposed [7]. Rather than requiring every full node to solve a mathematical puzzle, PoS randomly selects miners, called validators or forgers in PoS, to mine the next block. The probability of being chosen as validator linearly increases based on the amount of owned tokens.

The most popular smart contract platform Ethereum plans a hard fork to PoS in the near future [8].

2.2 Architectural Setup

Today's car have between 20 and 250 ECUs (electronic control units) [9]. Their functionality is to realize anti-blocking systems (ABS), lane departure warning, stability control, airbag deployment, wiper control, light control, adaptive cruise control, automatic parking, and many others. These ECUs are connected with bus systems, in most cases with CAN [10] (controller area network) buses. In some cases FlexRay is used instead of CAN. FlexRay [11] inherits all properties of CAN and provides higher bandwidth and real-time. Because of bandwidth limitations, in most cars several CAN bus systems are used (see Fig. 1). One ECU can be connected to more than one bus because it works with or provides messages on different buses. It is also possible that one ECU acts as a gateway and forwards messages to a different bus. Together, they transmit (considering a car with many features) more than 10,000 messages per second. These messages are used to give state information (like vehicle speed or button status) to other ECUs. This is usually done with cyclic messages. If other ECUs have to be informed immediately about a status change, an additional message is sent. It is possible that an ECU reacts on state changes. For example, the ECU performs a

certain action if the button value for "pressed" is received. To realize information redundancy for a better reliability, in some cases the action is only performed if the value has the required state for some time. In this time several messages are received. The cycle rate varies between 20 milliseconds and in rare cases 100ms depending on the kind of signal. With these cycle rates, more than 200 different types of messages may be sent. All ECUs connected to the bus system can read all messages but only consider information being relevant for their functionality.

3 Threat & System Model

In an autonomous vehicle, information on the vehicle itself (like speed and steering angle) and its environment perceived by appropriate sensors (like lane, traffic signs and position/movement of other traffic participants) is used for improving the algorithms which control the autonomous vehicle. That information can also be used to analyze accidents. It should not be possible to tamper with this information, because false information could lead to algorithms that cause accidents. Furthermore, the analysis of incorrect data impedes discovering the true reason in case of an accident.

We target an attacker that aims to veil its own responsibility for an accident, to blackmail the OEM (original equipment manufacturer), or to kill a certain car driver. To perform such an attack, the attacker may modify information. This can be done by using an existing wireless connection to exploit a vulnerability that can be used to access the internal bus system. Information can also be changed by replacing an ECU with a modified ECU that sends forged information to other ECUs. An attacker can also modify information by accessing the internal bus system using the diagnosis connector that exists in all vehicles being built in the last 15 years. The diagnosis connector is directly connected to the internal bus. Since diagnosis connectors usually also provide electric power, a device can be connected that establishes a wireless connection with direct access to the internal bus.

We assume that the attacker does not have the capabilities to perform a 51 %-attack as is typical for most blockchain-based systems based on majority voting because in our scenario the attacker has to replace more than half of the ECUs.

We also assume that the ECUs send messages with MACs (message authentication codes) to show that these messages come from a authentic device.

4 Architecture

In this section, we propose a blockchain-based architecture to combine safety and security for an application inside autonomous cars.

4.1 Blockchain Architecture

Our blockchain architecture uses the infrastructure consisting of ECUs and the connecting CAN buses (cf. Section 2.2). An ECU can add new transactions and

also works as a miner to add new blocks to the chain. ECUs that work as miners hold a copy of the blockchain. As transactions for the blockchain, we consider status messages that are relevant for safety or security, e.g. braking, steering commands etc. Therefore, manufacturers are able to define filters for types of messages that are safety or security critical. Each ECU implements those filter mechanisms to select relevant messages.

Since blockchain ensures integrity of stored data, our approach can be used as a blackbox for cars. Additionally, the blockchain consensus algorithms only allow to add validated transactions to the blockchain. With regard to safety, validated data is of high importance, e.g. for machine learning to improve algorithms for driver assistance or autonomous driving.

Currently, the performance that is needed for real-time operating on the blockchain is a big challenge. Since we use the CAN bus of a vehicle to broadcast all transactions and blocks, our architecture still allows reacting on the status messages before these are added to the blockchain.

For operating the blockchain of our architecture, the ECUs have to carry out different steps we describe in the following.

4.2 Step 1: New Status Message

As mentioned earlier, we only consider security and safety related status messages as new transactions for the blockchain. For those messages, we proceed in the following way: Each ECU holds a private key to sign its messages and a set of public keys from all other ECUs. Before broadcasting a transaction, an ECU signs the corresponding status message with its private key. Later, the signature can be verified by others using the related public key. The ECU adds the transaction to the set of transactions which are already stored in the ledger. Those are represented by the lifeline *Transactions*. Afterwards, the ECU broadcasts the transaction via the CAN bus on which other ECUs listen.

In case that the status message is not security or safety related, the ECU broadcasts it directly via the CAN bus.

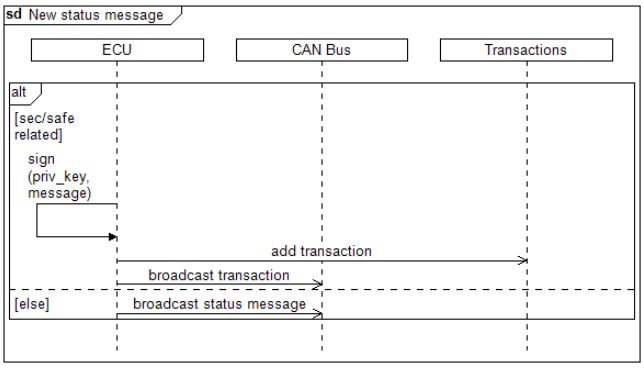

Fig. 2. New status message from ECU

The corresponding sequence diagram is given in Fig. 2.

4.3 Step 2: Verify Transaction

As mentioned before, other ECUs are listening on the CAN bus for new transactions. When an ECU receives a transaction, it needs to be verified prior to being accepted. The verification process consists of two steps: First, using the public key of the sender, the ECU verifies the signature of the message. In case that the verification fails, the message will not be further processed. Otherwise, the ECU checks the plausibility of the status message using reference values or specific algorithms. For example, a ECU can compare values of different sensors. If the verification succeeds, the ECU adds the transactions to the list of transactions to be added to the blockchain.

When reaching a certain threshold which is defined by a given number of transactions to be added to the blockchain, the ECU proposes a new block. It requests the last block and the list of transactions and calculates the corresponding hash for the new block. Last, the ECU adds the new block to its own copy of the blockchain and broadcasts via the CAN bus.

In case the verification fails, the ECU broadcasts an error. We show the corresponding sequence diagram in Fig. 3.

4.4 Step 3: Update Blockchain

Whenever a new block has been broadcasted, each ECU has to update its own copy of the blockchain. Before accepting a new block, the ECU has to verify the block itself and the contained transactions. For verifying a transaction, we use the procedure as described in the second step. When the transactions and the hash for the proposed block are valid, the ECU appends the proposed block to its copy of the blockchain. When a block is not valid, it is not processed further, and the ECU broadcasts an error message. We show the corresponding sequence diagram for the last step in Fig. 4.

Since each ECU now holds a redundant and validated copy of all status messages in form of transactions in the blockchain, all ECUs works on the same data set, and there is no single point of failure. The data set can be used for machine learning of algorithms, and for accident forensics. Assuming that a real-time processing of the blockchain is possible, the architecture can also be used to realize a MooN safety architecture. The blockchain validation process works with majority voting, and broadcasted error messages can be used to identify malfunctioning of equipment.

5 Discussion

In the following, we discuss the results we obtained from our proposed architecture.

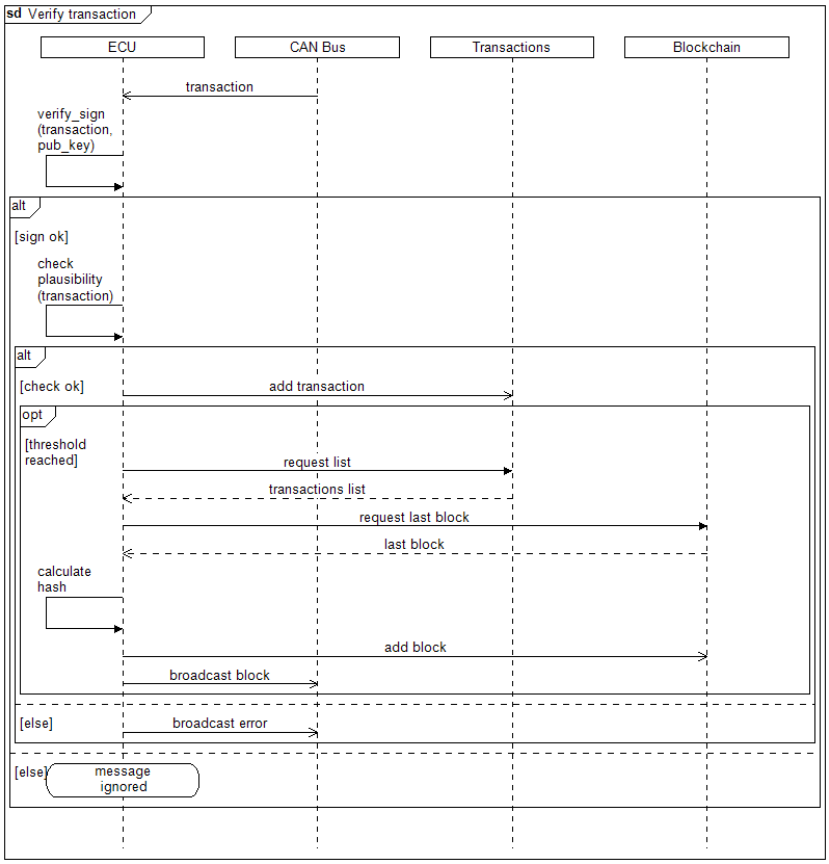

Fig. 3. Verify Status Message

5.1 Performance & Scalability

One of the main drawbacks of using blockchains inside vehicles is the increasing need for computational power. We suggest to focus on security and safety-relevant status messages and to combine a certain number of those messages into one block. The reduced number of transactions and block will limit the required computational power and required storage capacity.

Another approach can be to use *Trusted Platform Modules (TPM)* for cryptographic operations, e.g. hash calculation and creating Message Authentication Codes (MACs) as specified in the *TPM Automotive Thin Profile* [12].

For inserting a block into the blockchain, we suggest using *Proof-of-Stake*. In contrast to *Proof-of-Work*, which is for example used by Bitcoin, the required computational power is limited, because the different ECUs do not compete in proposing new blocks. Calculating a hash for a new block does not require to solve cryptographic puzzles.

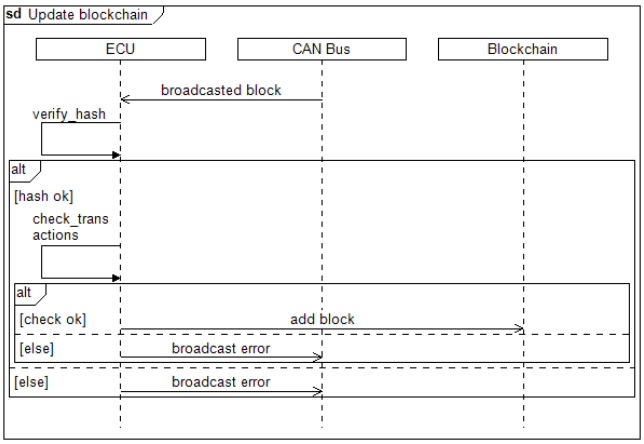

Fig. 4. Add block

Last, we suggest using our framework for future autonomous vehicles. Due to the various sensors and real-time algorithms which processes the measured values, those vehicles provide higher storage capacity and more computational power than current vehicles.

5.2 Safety & Security

Our blockchain-based framework focusses on preserving integrity for security and safety-related status messages which are broadcasted via the CAN bus in vehicles. The processes we described in Section 4 include plausibility checks and message signatures for those messages. Those checks ensure that only valid data is further processed by the blockchain and other ECUs. Furthermore, the history of received messages provided by the blockchain ensures that those messages cannot be manipulated by compromising a single ECU or by injecting malicious status messages. To manipulate the blockchain, it is necessary to perform a 51% attack which means that an attacker has to compromise more than half of ECUs.

5.3 Privacy

Since blockchain-based architectures store a history of messages which cannot be altered or partially deleted anymore, it raises the need for considering privacy aspects, as well. Autonomous cars process sensitive data like GPS coordinates. Since we only use blockchain for internal vehicle communication, those data is not processed to external entities. Nevertheless, in case of accidents or for maintenance reasons, the responsible third party will take a copy of the blockchain, at least partially. Therefore, it is necessary to define privacy policies for the blockchain or to restrict access to the data.

6 Related Work

Cebe et. al. [3] describe a blockchain approach for forensic crash data investigation. For this, e.g. data from traffic lights, tire pressure, wiper state, and vehicle speed are used. The authors make use of an externally shared ledger which does not allow to verify the vehicle's data.

Oham et. al. [13] also propose a blockchain where the identity of the validators or even the participants is whitelisted (called permissioned blockchain) in the vehicle's environment to support the liability of entities in case of accidents. The focus of the proposed architecture lies on liability attribution for entities such as car manufacturers in case of an accident.

Ugwu et. al. [4] state that neither with the blockchain in [3] nor with the blockchain in [13] the proof of vehicle state is possible. In this paper, the authors suggest proving the state of the sensors in a smart vehicle by tracking the changes in the state of the smart vehicle's sensors and to record changes in the blockchain. The approach still requires a trusted authority for maintaining the externally shared ledger. Additionally, the approach requires a permanent connection to this authority in case of an accident to transmit the vehicle's state information.

Dorri et al. [14] propose a distributed blockchain-based framework to address security and privacy in interconnected smart vehicles. The authors state several use cases for their framework, e.g. car-sharing services or central payment for the power consumption of electric vehicles. Privacy-related information such as location data is still stored locally inside the vehicle and is not transferred to the public blockchain. Therefore, the benefits of the approach for accidents forensics is limited. The locally stored data can still be modified.

There is a new initiative called *Mobility Open Blockchain Initiative (MOBI)*[1] in which several well-known OEMs and suppliers take part. The aim of this group is to elaborate on blockchain-based solutions in the vehicle's environment.

To the best of our knowledge, there is no related work considering blockchain architectures inside vehicles. Our blockchain-based architecture may be considered as an extension to verify the vehicle's data for the above-mentioned works.

An alternative to using a blockchain is a realization with cryptographic methods as described in CC PPs (Common Criteria Protection Profiles) being already available for the Digital Tachograph [15] consisting of a Smart Card (Tachograph Card), the External GNSS Facility, the Motion Sensor, and the Vehicle Unit. This PP shows the potential structure for the realization of a device collecting information for insurances to collect data relevant for accident forensics. Advantages of using a blockchain are the synergies to safety, the redundant storage and that for a successful attack several ECUs have to be compromised.

[1] https://www.t-systems.com/at/de/newsroom/blog/automotive/automotive/blockchain-technologie-fuer-fahrzeuge-823460

7 Conclusion

In this paper, we proposed a blockchain-based architecture for autonomous vehicles to combine safety and security aspects. Using a blockchain ensures the integrity of security and safety-relevant status messages of electronic control units (ECUs). The recorded history of messages can be exploited for accident forensics and machine learning.

A distinguishing feature of our approach is that we store the blockchain internally and make use of the existing CAN bus infrastructure. Existing approaches only describe the usage of external blockchains which do not ensure integrity of data inside vehicles.

Furthermore, using internal storage solutions for the blockchain improves the privacy of the car owner and driver since the data will not be made available for third parties. We described a high-level algorithm in detail to perform message authentication and plausibility checks in our framework that ensures that ECUs add only valid data to the blockchain.

As future research directions, we will implement a prototype of our proposed architecture to assess its performance and therefore its scalability. The obtained results will further be used to develop a suitable consensus algorithm.

In addition, our approach complements orthogonal work on accident forensics. Making parts of our blockchain accessible to external authorized third parties will improve several use cases, e.g., insurances can benefit from the integrity of data provided by the car. Therefore, we will elaborate on combining external blockchain solutions (cf. Section 6) with our approach.

References

1. Miller, C., Valasek: Securing self-driving cars (one company at a time). http://illmatics.com/carhacking.html (August 2018)
2. Tesla: Tesla Blog - What we know about the last weeks accident. https://www.tesla.com/de_DE/blog/what-we-know-about-last-weeks-accident Accessed: 15th January 2019.
3. Cebe, M., Erdin, E., Akkaya, K., Aksu, H., Uluagac, S.: Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. IEEE Communications Magazine **56**(10) (OCTOBER 2018) 50–57
4. Ugwa, M.C., Okpala, I.U., Nwakanma, C.I.: A tiered blockchain framework for vehicular forensics. IJNSA **10**(5) (2018)
5. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf (2008)
6. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper **151** (2014) 1–32
7. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. In: Advances in Cryptology – CRYPTO 2017, Springer International Publishing (2017) 357–388
8. Cavicchioli, M.: When will ethereum's proof of stake arrive? https://cryptonomist.ch/en/2018/11/01/when-will-ethereums-proof-of-stake-arrive/ (November 2018)
9. Ebert, C., Jones, C.: Embedded software: Facts, figures, and future. Computer **42**(4) (April 2009) 42–52
10. International Organization for Standardization - ISO Geneva, Switzerland: Road vehicles – Controller area network (CAN) series. (2015)
11. Regler, R., Schlinkheider, J., Maier, M., Prechler, R., Berger, E., Pröll, L.: Intelligent electrics / electronics architecture. ATZextra worldwide **15**(11) (Jan 2010) 246–251
12. TCG: TCG TPM 2.0 Automotive Thin Profile. https://trustedcomputinggroup.org/resource/tcg-tpm-2-0-library-profile-for-automotive-thin/ (3 2018)
13. Oham, C., Kanhere, S., Jurdak, R., Jha, S.: A blockchain based liability attribution framework for autonomous vehicles. (02 2018)
14. Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R.: Blockchain: A distributed solution to automotive security and privacy. IEEE Communications Magazine **55**(12) (Dec 2017) 119–125
15. European Commission DG JRC: Common Criteria Protection Profile – Digital Tachograph. https://www.commoncriteriaportal.org/search/?cx=016233930414485990345(May 2017)